OPTIMIZING DISTRIBUTED SENSOR PLACEMENT FOR BORDER PATROL INTERDICTION USING MICROSOFT EXCEL

THESIS

Adrian Patrascu, Capt, USAF

AFIT/GOR/ENS/07-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/07-21

OPTIMIZING DISTRIBUTED SENSOR PLACEMENT FOR BORDER PATROL INTERDICTION USING MICROSOFT EXCEL

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Adrian C. Patrascu, BA

Capt, USAF

April 2007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GOR/ENS/07-21

OPTIMIZING DISTRIBUTED SENSOR PLACEMENT FOR BORDER PATROL INTERDICTION USING MICROSOFT EXCEL

Adrian C. Patrascu, BA
Capt, USAF

Approved:

____________________________ ____________
James T. Moore, Ph.D. (Chairman) Date

____________________________ ____________
John O. Miller, Ph.D. (Member) Date

AFIT/GOR/ENS/07-21

Abstract

The purpose of this research was to develop a method for sensor placement on a Border Patrol interdiction network. Specifically, this thesis sought to develop a proof of concept model using Microsoft Excel, with some add-on capabilities, to optimize the probability of detecting intruders who have already breached the border through the placement of electronic sensors on a network. A model was developed which maximizes the probability of detecting intruders by optimizing the build-up of a distributed sensor network subject to a budgetary constraint. Several different optimization algorithms were developed for use with the model. All were tested and their results were analyzed revealing two very promising sensor placement methods for optimizing sensor coverage on a network.

Due to its ease of use and ability to run in Microsoft Excel, it is believed that the model developed in this research can also be used in a number of military applications where border security is necessary.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. James Moore, and my reader, Dr. John O. Miller for their guidance and support throughout the course of this thesis effort. Their combined insight and experience were greatly appreciated. I would, also, like to thank Captains David Kerns, Robert Koo, Yuri Taitano, Ryan Caulk, and Kevin Reyes for their help and friendship throughout my time at AFIT. Finally, I would like to thank my family for their understanding and support.

Adrian C. Patrascu

Table of Contents

Page

List of Figures

List of Tables

OPTIMIZING DISTRIBUTED SENSOR PLACEMENT FOR BORDER PATROL INTERDICTION USING MICROSOFT EXCEL

I. Introduction

Background

The United States of America is not only the world's sole true superpower, but also its most hospitable host. Founded by immigrants for immigrants, the U.S. takes in more of the world's poor and downtrodden than any other nation. In fact, as of 2006 "the United States accept[ed] more legal immigrants as permanent residents than the rest of the world combined" (9). In America, a poor immigrant can become a CEO of a major corporation or the governor of the most populous state in the nation. A study from Duke University found that

> 25 percent of technology and engineering companies started from 1995 to 2005 had at least one senior executive - a founder, chief executive, president or chief technology officer - born outside the United States. (10)

While legal immigration has always been a great boon to the United States, recently there has been an alarming increase in illegal immigration. The government of the United States has been unable or unwilling to stop the flow of smugglers and illegal aliens across its borders. Over the years, millions of people have entered the country illegally; mostly across the southern border. In fact, as of 2004, there were an estimated 12 to 20 million illegal aliens in the United States. (11)

In the past, the internal debate in the United States for and against illegal immigration has centered mostly on economics. Businesses have enjoyed the cheap labor provided by illegal immigrants while workers and union groups have decried the mass hiring of illegal aliens

(especially in sectors such as agriculture and construction) as a cheaper alternative to American citizens and legal immigrants.

However, with the rise of the radical Islamic movement in the 1980s and 90s, and the subsequent terrorist attacks of September 11, 2001, overlooking illegal border crossings is no longer an option. Nowadays, it is not just drug smugglers and poor people looking for work that are of concern. There is now a very real threat of terrorists using our porous borders to infiltrate and attack our country. The nation must protect its borders in order to protect its citizens from the very real threats which face them. For this reason, the U.S. government must take a careful look at all people that are entering this country and do its utmost to prevent unauthorized entry into the United States. However, despite the ever-present threat, it is estimated that an average of half a million people cross the U.S.-Mexico border illegally each year and the U.S Border Patrol catches only 1 in 4 border crossers. (5; 12:14)

Recently, in an effort to, among other things, improve the management of its borders, the U.S. government has undertaken a large reorganization effort. On March 1, 2003, as part of a realignment effort after September 11, 2001,

> the Department of Homeland Security was established. It was and is the largest reorganization of our Federal Government in over 50 years. As part of the Department of Homeland Security reorganization, U. S. Customs and Border Protection – CBP – was created by unifying all frontline personnel and functions with law enforcement responsibilities at our nation's borders, that is, at all 300 plus ports of entry of the United States – land, sea and air - and the areas in between the official ports of entry. (13:2)

After September 11, 2001, the new priority mission of the Border Patrol became to "prevent terrorist and terrorist weapons from entering the United States." Of course the traditional mission of the Border Patrol, to prevent "illegal aliens, smugglers, narcotics, and other contraband, from entering the United States," remains, and in fact coincides with this new priority. (13:2)

Unfortunately, creating a new bureaucracy, with central leadership, while a commendable idea, does little to actually improve the desperate situation at our borders. For example, in January of 2005, while pursuing three suspected drug smugglers in SUVs,

> three Hudspeth County [Texas] deputies and at least two Texas Department of Public Safety troopers squared off against at least 10 heavily armed men from the Mexican side of the Rio Grande. U.S. officials who pursued three fleeing SUVs to the Mexican border saw what appeared to be a Mexican military Humvee help one of the SUVs when it got stuck in the river…When that didn't work, a group of men dressed in civilian clothes started unloading what appeared to be bundles of marijuana from the SUV, and the stuck vehicle was then torched… A second SUV had a flat tire and was left behind in the United States and its occupant ran across the border. (14)

Again, in November 2005, U.S. border patrol agents attempted to pull over a suspect truck on Interstate 10 in Texas. The driver, rather than pulling over, exited the freeway and fled south towards the Rio Grande and the Mexican border. While attempting to cross the Rio Grande, the truck got stuck and the driver got out and fled into Mexico on foot. The border patrol found 3 tons of marijuana in the truck and called for reinforcements from the Texas State Troopers. Shortly after the troopers arrived and the officers started emptying the truck, the driver, who had fled into Mexico, returned with an armed militia and a bulldozer to pull the truck out. The U.S. Border Patrol and the Texas State Troopers, outnumbered by the heavily armed Mexican militia, were forced to allow the Mexicans to retrieve the truck, still two thirds full of marijuana, and take it back into Mexico. (15)

In yet another incident "Chief Deputy Mike Doyal of the Hudspeth County [Texas] Sheriff's Department said that Mexican army personnel had several mounted machine guns on the ground more than 200 yards inside the U.S. border" (14). Even more terrifying is the recent Department of Homeland Security report stating that Mexican troops (or armed paramilitary forces dressed in Mexican army uniforms) have entered the U.S. at least 216 times in the past 9 years. (16)

Because of the serious nature of the situation on its southern border, it has become clear that the U.S. must do more to protect its borders and its citizens. In addition to hiring more border agents (or putting the National Guard on the border) and providing them with proper training and equipment, the US has the ability to help its border agents by using a number of high-tech devices to detect and track illegal crossings into our country. Smart fences, multiple types of cameras, radar towers, and seismic sensors linked through a system-wide wireless communication network can be used to detect and track subjects. Making use of these new technologies, border agents will be able to perform their jobs better, more efficiently, and with higher success rates than ever before.

Problem Statement

The United States has approximately 7000 miles of border with Canada and Mexico. Of these 7000 miles of border, most of the Canadian, and large parts of the Mexican border, are almost completely unprotected.

> The U.S. Border Patrol has 20 sectors responsible for detecting, interdicting and apprehending those who attempt to illegally enter or smuggle people, including terrorists, or contraband, including weapons of mass destruction, across U.S. borders between official ports of entry. (17)

For example, the El Centro Sector covers the Riverside and Imperial counties in southern California. (17)

Although the Border Patrol does have agents assigned to each section of the border, the available manpower, given the magnitude of the task, is insufficient to cover every possible entry point. In addition, up until the present day, the use of technology on the border to help with interdiction efforts has been limited. While technology is being used on some sections of the border presently, what is needed is a systematic effort to implement technological solutions into the areas of the border patrol effort where they are most effective. By introducing technological

solutions on the US border, especially in the areas of detection, tracking, and communications, it is possible to help agents do their jobs much more efficiently and effectively. It is also much cheaper to add a technological infrastructure to help agents do their jobs, than it is to hire the large number of additional agents that will otherwise be required. (18)

Research Objective

The primary objective of this research is the development of a model optimizing the placement of electronic sensors on a border network given a pre-determined budgetary constraint. The model is capable of handling multiple sensor types which are placed together as packages. Also, some sensors operate during daytime, others operate during nighttime, and yet others operate both during daytime and nighttime. The probability of an intruder being detected, by each sensor type, is calculated for each node in the network. Then, the probabilities for each sensor type are combined using the assumption of independent probabilities. A separate probability of detection at each node is calculated for daytime and nighttime and both (by taking the average of the daytime and nighttime probabilities). The model then uses several techniques to place sensors at nodes in order to maximize coverage (probability of detection) on the network.

Research Focus

The research is focused on creating a proof-of-concept model for placement of a distributed network of unattended electronic sensors in order to maximize the probability of detecting intruders. The model maximizes the probability of detecting illegal aliens using heuristic methods to place electronic sensors creating an interior surveillance network capable of detecting intruders after they have already breached the border. The model will not account for technology placed on the border itself (such as smart fences). As an additional requirement, the

model will be easy to implement and modify giving the user the ability to quickly make changes and re-run the model in order to adapt to changing requirements. Microsoft Excel is the software of choice for this research because of its high world-wide market share. In fact, Microsoft "owns more than 90 percent of the office productivity application market" through their Office software suite, which contains Excel (19). Therefore, it is not unreasonable to assume that, in any organization, there is at least one person who knows how to use Excel; making the model much more likely to be used.

Overview

The remainder of the document has a review of prior interdiction related research as well as the software which will be required to complete this research. This is presented in Chapter 2.Then, the model, which is object of this research, is developed and tested. Finally, future research recommendations are made.

II. Literature Review

National Border Patrol Strategy

The Border Patrol's strategy for protecting the national borders, as stated in the National Border Patrol Strategy of 2004, consists of the following five objectives:

1. Establish substantial probability of apprehending terrorists and their weapons as they attempt to enter illegally between the ports of entry. (13:7-11)

2. Deter illegal entries through improved enforcement. (13:7-11)

3. Detect, apprehend, and deter smugglers of humans, drugs, and other contraband. (13:7-11)

4. Leverage "Smart Border" technology to multiply the effect of enforcement personnel. (13:7-11)

5. Reduce crime in border communities and consequently improve quality of life and economic vitality of targeted areas. (13:7-11)

The Border Patrol has identified four approaches they will use to achieve the outlined objectives:

1. A more flexible, well-trained, nationally-directed Border Patrol. (13:7-11)

2. Specialized teams and rapid-response capabilities. (13:7-11)

3. Intelligence-driven operations. (13:7-11)

4. Infrastructure, facility, and technology support. (13:7-11)

Ninety percent of arrests made by the Border Patrol each year occur along the 2000 mile long U.S. border with Mexico.

> The Border Patrol has experienced success in gaining operational control of the [Southern] border in some of the highest trafficked areas, such as San Diego [CA], El Paso [TX], and McAllen [TX]. However, many other areas along the southwest border are not yet under operational control, and the daily attempts to cross the border by thousands of illegal aliens from countries around the globe continue to present a threat to U.S. national security. (13:5)

The Border Patrol has identified the following strategies for controlling the U.S.-Mexico (Southern) border:

1. Deter or deny access to urban areas, infrastructure, transportation, and routes of egress to smuggling organizations through checkpoints, intelligence-driven special operations, and targeted patrols; (13:15-16)

2. Expand control through increased and more mobile personnel and improved air and ground support; (13:15-16)

3. Increase rapid response capabilities; (13:15-16)

4. Continue and expand the appropriate mix of improved infrastructure and technology; (13:15-16)

    a. Sensing systems, Remote Video Surveillance and Sensing (RVSS) cameras, air support, and Unmanned Aerial Vehicles (UAVs) (13:15-16)

    b. Radiation detection equipment (13:15-16)

    c. Improved communication infrastructure (Land Mobile Radio, cellular coverage, satellite communication capabilities) (13:15-16)

    d. Remote access to national law enforcement databases through the use of mobile computing solutions (13:15-16)

Network Interdiction

Network Interdiction involves a network user trying to utilize a network to optimize the movement of goods and information, while a network interdictor attempts to stop or reduce the movement of material and information through the network. From a military perspective, interdicting is generally modeled by destroying the nodes of a network or reducing their effectiveness below a predetermined threshold. However, for border interdiction, the goal is to optimize coverage over a given network in order to improve the success rate of interdiction efforts. Destroying the nodes of the network is generally not an option.

Shortest Path Network Interdiction.

In their article, Eitan and Wood develop a new method for maximizing the shortest-path between two nodes in a network. If the interdictor had an unlimited budget, he would simply solve the cut-set problem and destroy all of the designated arcs thus making it completely impossible for the user to move anything across the network. Of course, this is rarely the case. In reality, there will be budgetary constraints which the interdictor must follow. Therefore, while the interdictor may not be able to completely cut the network, he can maximize the length of the shortest path. The object is to destroy (or lengthen) a select number of arcs in order to optimize the disruption to the network under the budgetary constraints placed on the interdictor. (20)

The shortest-path network interdiction problem can be solved using a branch and bound plus linear programming relaxation approach. However, this method can be very time consuming, especially when dealing with large networks. Additionally, in the military realm, the need for solutions is often time sensitive. For this reason, and others, the authors have developed an algorithm that improves on the efficiency of a linear relaxation solution. (20)

Eitan and Wood started by formulating the "Maximizing the Shortest Path" (MXSP) problem as a Mixed Integer Programming (MIP) problem. They also developed four separate decomposition methods which solve problems quicker than the traditional branch-and-bound linear programming approach. On its own, Benders' decomposition performed quite poorly, but with the addition of "supervalid inequalities", it showed significant improvements in computational efficiency. (20:97)

The "Supervalid Inequality" (SVI) introduced in the article can be viewed as a generalized version of the "standard valid inequality" (or "cut"). However, whereas the standard valid inequality would not reduce the number of feasible solutions, the SVI does indeed reduce

the number of feasible solutions. Furthermore, the feasible solutions are reduced in such a way as to guarantee that the optimal solution is not removed (unless the incumbent solution is also the optimal). (20:100)

Out of the four decomposition algorithms developed in the article, two work quite well with the MXSP problem. In fact, all of them offer an increase in efficiency over the classic branch-and-bound LP. However, the most intriguing aspect of this article is the second decomposition algorithm because it can be generalized to other network and system interdiction problems. Indeed, the authors claim this algorithm is already being used to solve a "tri-level system defense problem" in order to "harden a road network against attack." (20:110)

The SVIs developed in this article proved to be a very useful tool. Using SVIs, optimality was determined significantly faster than with Benders' decomposition. The main shortcoming of all of these algorithms is reduced flexibility to one degree or another. SVIs are an effective tool for more efficient and faster solutions, but they can only be used for a reduced set of problems. If time is not of the essence, it may be easier and simpler to employ the classic branch and bound plus LP relaxation technique which will theoretically solve all IPs and MIPs eventually. (20)

LP Optimization.

Pulat [2005] develops a mixed integer linear program which optimizes border interdiction in the Yuma sector of the U.S.-Mexico border. He studies scenarios where the intruder is traveling in a vehicle and scenarios where the intruder is traveling on foot. The scenarios are further divided into the case where the intruder knows the U.S. Border Patrol's positions ahead of time, versus the case where the intruder is not pre-aware of these positions. Pulat also makes a "distinction between actions that can only lead to detection [sensors, helicopters] and action that

can also lead to capture in addition to detection [road patrols, checkpoints, remote observation posts]." (21:39)

Pulat uses a network representation of nodes, arcs, and centers of land parcels overlaying a satellite map of the Yuma, Arizona border area. He uses open source information from the Border Patrol and identifies all candidate defensive actions based on the location of checkpoints, road patrols, off-road operations, remote observation posts, and electronic sensors. He also identifies intruder actions and creates a "Two-Sided Mixed Integer Optimization Model to Minimize Maximum Probability of Escape" (21:25). Using a number of different scenarios, Pulat identifies critical road segments and land parcels to be defended and studies the "effects of employing different types of assets and strategies on the infiltration patterns." (21:39)

Continuous Network Interdiction.

Washburn [2006] develops a network interdiction model for economic networks with indefinite time outlooks. This model seeks to minimize the fraction of product that makes it from its origination point to its destination point without being interdicted. The model is developed as a two-person zero-sum game. He also explores the consequences of allowing the interdictor to sell confiscated goods. This not only increases the interdictor's budget, leading to a larger interdiction effort, but also depreciates the commodity making it harder for the shipper to make a profit. "This leads to a Nash equilibrium where the shipper's quantity shipped is in equilibrium with the interdictor's budget for interdiction." (22:1)

Game Theory Approach.

Washburn and Wood [1995] develop a game theory approach to network interdiction. The game takes place on a network of nodes and arcs with one evader and one interdictor. For each arc in the network, a constant probability of detection is determined beforehand. Then,

while the evader determines a "path-selection" strategy minimizing the probability of detection, the inspector determines an "arc-inspection" strategy maximizing the probability of detection. The authors show that this type of problem can be solved using standard network flow techniques. They also discuss problems with "unknown origins and destinations" as well as "multiple interdictors and evaders." (23:243)

Sensor Placement

Remote sensing technologies have the potential of greatly reducing the number of personnel needed for border patrol while at the same time increasing the probability of detecting and capturing intruders. While the border patrol has been using a limited number of electronic sensors and other devices for a number of years, they do not have an integrated electronic network of sensors designed to detect, track, and aid in the capture of illegal aliens and smugglers.

Sensor Placement Algorithms for Effective Coverage.

Dhillon and Chakrabarty "present two algorithms for the efficient placement of sensors in a sensor field." (24:1609) Both algorithms are

> aimed at optimizing the number of sensors and determining their placement to support distributed sensor networks. The optimization framework is inherently probabilistic due to the uncertainty associated with sensor detections. The proposed algorithms address coverage optimization under the constraints of imprecise detections and terrain properties. These algorithms are targeted at average coverage as well as at maximizing the coverage of the most vulnerable grid points. The issue of preferential coverage of grid points (based on relative measures of security and tactical importance) is also modeled. (24:1609)

For both algorithms, it is assumed that

> the probability of detection of a target by a sensor varies exponentially with the distance between the target and the sensor. A target with distance d from a sensor is detected by that sensor with the probability $e^{-\alpha d}$. The parameter α can be used to model the quality of a sensor and the rate at which its detection probability diminishes with distance. (24:1610)

For every set of points i and j in the sensor field, two probability values are assigned: p_{ij}, which denotes the probability that a target at point j is detected by a sensor at point i, and p_{ji}, which denotes the probability that a target at point i is detected by a sensor at point j. The probabilities p_{ij} and p_{ji} are symmetric in most cases but can differ in the presence of obstacles.

Dhillon and Chakrabarty's first algorithm (MAX_AVG_COV) attempts to maximize the average coverage of the grid points, while their second algorithm (MAX_MIN_COV) attempts to maximize the coverage of the grid point which is least effectively covered; that is the grid where, if located, a target would have the least probability of being detected. Dhillon and Chakrabarty test the two algorithms, on an 8 by 8 grid, against each other as well as random an uniform placement of sensors. They conclude that the MAX_MIN_COV algorithm produces superior results, i.e. they achieved the best probability of detection (coverage) using this algorithm. Furthermore, they discuss continued research which would include minimum and maximum ranges for each sensor. (24:1614)

Sensor Placement Algorithm for Minimalistic Grid Coverage.

Dhillon, Chakrabarty, and Iyengar present

> a resource bounded optimization framework for sensor resource management under the constraints of sufficient grid coverage of the sensor field. The proposed theory is aimed at optimizing the number of sensors and determining their placement…The proposed algorithm addresses optimization under constraints of imprecise detections and terrain properties. The issue of preferential coverage of grid points (based on relative measures of security and tactical importance) is also modeled. (25:1)

For every set of points i and j in the sensor field, two probability values are assigned: p_{ij}, which denotes the probability that a target at point j is detected by a sensor at point i, and p_{ji}, which denotes the probability that a target at point i is detected by a sensor at point j. The probabilities p_{ij} and p_{ji} are symmetric in most cases but can differ in the presence of obstacles.

The algorithm uses an iterative "greedy heuristic" to determine the best placement of a single sensor one at a time. At every iteration, the algorithm adds one sensor and calculates the new probabilities for the entire grid. It also keeps track of improvements from previous iterations. The algorithm continues placing sensors until the miss probability for each point is smaller than the maximum permitted value. Preferential coverage areas in the grid can be implemented by lowering the maximum miss probability for preferred points and thereby forcing a higher probability of detection in those areas. Also, the algorithm

> makes the implicit assumption that sensor detections are independent, i.e. if a sensor detects a target at a grid point with probability p1, and another detects the same target at a grid point with probability p2, then the miss probability for the target is (1-p1)(1-p2). (25:4)

The algorithm presented by Dhillon, Chakrabarty, and Iyengar adds one sensor at a time to the grid until certain preset conditions are met. It is intended to determine the minimum number of sensors needed to meet the preset requirements. It does not backtrack in order to find the optimum placement of sensors at each iteration.

Sensor Technology

There are many useful sensor technologies which can be employed by the Border Patrol for intruder detection. Some of them are discussed below.

Cameras.

There are a large number of camera systems and technologies available from various defense-focused vendors. These include the more traditional daylight cameras, low-light level cameras, and infra-red (IR) cameras, as well as the newer and more sophisticated Forward Looking Infra-Red (FLIR) and Range-Gated cameras (26:1). FLIR cameras are thermal imaging cameras. Unlike traditional IR cameras, FLIR cameras do not require IR illuminators, which make it almost impossible for intruders to spot them. Unfortunately, FLIR cameras do have some

significant drawbacks as they do not work well in adverse weather conditions and they can be evaded by using techniques which minimize heat signatures (26:7). Range-Gated cameras make use of lasers and other technologies to literally see through snow, rain and fog at any time of the day or night (26:8). Two examples of camera systems are included below, but many others are available. Also note that, as technology continues to improve, the included examples will be outdated.

The GVS1000 (see Figure 1) is a "long-range active-infrared day/night surveillance system." It delivers 1.2 kilometers of "classification level" surveillance in complete darkness. It also contains integrated software which can "classify, recognize, and/or identify" targets. (1)

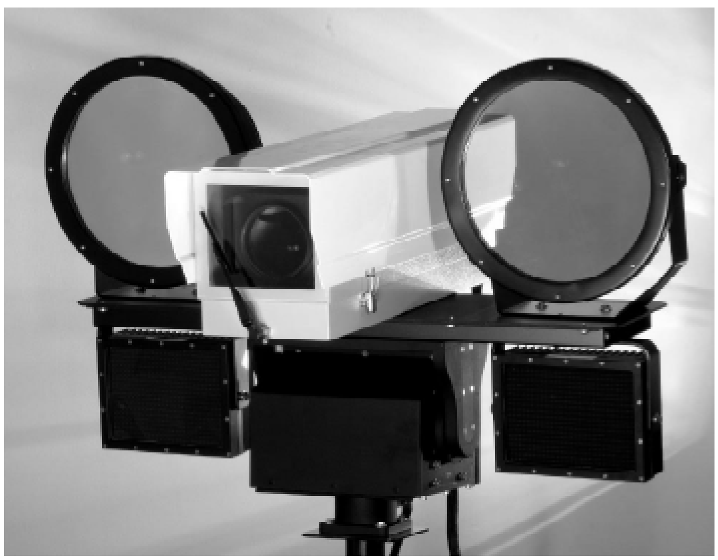


Figure 1. GVS1000 Long-Range Surveillance System (1)

The Axsys ExtremeXS thermal imaging camera (see Figure 2) is a "rugged camera with extensive detection capabilities." The ExtremeXS can detect a human sized target at up to 4.5 kilometers distance in less than ideal conditions. (2)

Figure 2. Axsys ExtremeXS Thermal Imagery Camera (2)

Ground Surveillance Radar.

The Motorola Modular Surveillance Radar is a man portable radar system capable of detecting a single person sized object up to 3 miles away. It can detect a small vehicle up to 7 miles away, and a larger vehicle up to 12 miles away. Groups of vehicles or people improve detectability. The MSR provides target location accuracy of 15 meters in range, and .6 degrees in azimuth. The radar system has been mounted to vehicles, trailers, and fixed site towers, and has been used operationally since 1990. The radar system provides wide area surveillance. When a daylight/infrared camera system is used in combination with the radar, target identification is possible. The radar can be remotely controlled by radio link, or long haul RS-232 lines. (3)

Additionally, Dragoon Technologies has

developed a modern map based application for detected target display. The application can be used to steer additional sensors and accepts GPS input for mobile applications. The radar system utilizes mil-spec construction and operates on 24VDC." (3)

Figure 3 shows the Motorola MSR-20 Ground Search Radar mounted on a tower along with a video camera and infra-red sensor.

Figure 3. Motorola MSR-20 Ground Search Radar (3)

Seismic Sensors.

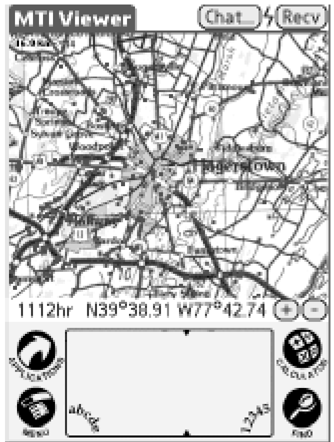
There are a number of promising seismic sensing technologies which can be used for border security. Maier [2004] developed a seismic intrusion sensor technology which uses buried fiber optic cables, lasers, and piezoelectric transducers, to detect and locate walking intruders at distances up to 2 kilometers away (37). In 2002, the U.S. Army Corps of Engineers tested a different approach to a seismic sensing system. This system used nodes made up of six sensors placed in a circular area with a 6 meter diameter. The test concluded that seismic sensors were effective at distances up to 1 km away under simulated battlefield environments. (38)

Palm PDA Based Intelligence Distribution.

Getting sensor data collected, processed, and distributed to officers in the field can be a lengthy process if it involves human-in-the-loop interactions. Dragoon Technologies has

> developed two applications [Figure 2] for PDA computers that put MTI data, video and freeze frame imagery onto a computer slightly larger than a deck of playing cards. The screens are sunlight viewable and the form factor is soldier/operative friendly. Communications to these devices is currently available as RS-232, TCP/IP, WiFi, Bluetooth, and cellular telephone. The links all include the ability to send bi-directional data to include GPS position of the Palm PDA to a server and chat messaging. Live

streaming video is now available as well. The PDA represents the future of intelligence distribution to scaled, mobile devices. (4)



MTI data as viewed on Palm PDA

Imagery "zoomified" and viewed on Palm PDA

Figure 4. Palm PDA based intelligence distribution (4)

PDA devices have the potential to provide border agents in the field with near instantaneous information enabling them to track and capture intruders with unprecedented ease.

Software

In addition to Microsoft Excel, a few other software programs are needed to create the proof-of-concept network. Since the Border Patrol does not make their maps and mapping software available to the public, Google Earth is used as the base map for the network.

Google Earth.

Google Earth is a free virtual globe mapping program originally developed by Keyhole Inc., but later purchased and distributed by Google Inc. It provides satellite images of the entire world with overlaid road maps and, in some places, 3D terrain and building models. The program also allows users to create and store their own points of interest called "place marks". (6)

GEPath.

GEPath is a freeware program “developed to make paths and/or draw circles and polygons with place marks saved by Google Earth.” It parses Google Earth “kml” files (kml files are files written in extensible markup language and used by Google Earth to show user specified information) and retrieves place mark information such as the place mark’s name, latitude, and longitude.

> The data can also be typed into the application or pasted/exported to the clipboard. Files generated by GE-Path are exported to Google Earth. This application calculates distances, bearing and area. (7)

Frontline Systems OptQuest Solver.

The Excel Solver allows the user to optimize a given objective function based on a set of changeable cells (variables) and a set of constraints. Frontline Systems is the company which developed the Excel Solver for Microsoft. However, the Excel Solver is limited in its scope. It has a maximum capacity of 200 variables and constraints for linear models and can only solve a limited number of non-linear models. Frontline Systems offers the Premium Solver Platform and a number of “field installable engines” to extend the capability of the Excel Solver. Specifically, the OptQuest solver (one of the field-installable engines) “employs metaheuristics such as tabu search and scatter search to solve nonsmooth optimization problems of up to 5,000 variables and 1,000 constraints. It also supports integer variables.” While not guaranteeing an optimal solution, the OptQuest Solver “finds remarkably good solutions with unprecedented speed.” (27)

In the next chapter, Border Patrol interdiction and the need for innovations is discussed in detail. After explaining the need for technological innovation for border security, a model is developed optimizing the placement of electronic sensors in order to maximize the probability of detecting intruders.

III. Methodology

In this chapter, the traditional approach to border patrol is discussed along with the need for a new approach. Then, a model is built for placement of distributed sensors on a network with the goal of maximizing the probability of detecting intruders. This model is intended to be the first part of the overall strategy of creating a new technological approach to border patrol.

Traditional approach

Traditionally, border protection has been a very manpower intensive job. The job requires many border patrol agents in vehicles, on horseback, or on foot to patrol areas searching for intruders. Intruder detection can also be performed by helicopter patrols, but, while helicopters greatly improve speed and the probability of detection, they are expensive to purchase, fly, and maintain. Once intruders are detected, the patrol agents must change tasks and attempt to apprehend the intruders.

Another traditional method for border protection is the interior checkpoint. The Border Patrol uses both permanent and temporary immigration checkpoints where all vehicle traffic is stopped in order to detect and apprehend illegal aliens, drugs, and other illegal activity. The permanent checkpoints are generally located on national roads and interstate freeways, while temporary checkpoints, called "tactical checkpoints," are located on smaller arterial and rural streets with traffic volumes as small as a few hundred vehicles per day (5). The 2005 Government Accountability Office report on immigration checkpoints claims that "while changing locations of tactical checkpoints would appear to offer the potential element of surprise… the border patrol [claims] that the smugglers of aliens and contraband…use cell phones and communications equipment to alert confederates of the presence of checkpoints within minutes of their being relocated" (5:23,24). However, despite the fact that smugglers have

become increasingly more sophisticated in their use of technology, there is sufficient reason to believe that checkpoints make up a useful part of a multi-layered border protection strategy. For example, in 2004, the Border Patrol's Southwest interior checkpoints used 10 percent of the region's border patrol agents, contributed to 8 percent of the total number of apprehensions, and 31 percent of marijuana and 74 percent of all cocaine seizures. (5:29,30)

Figure 5. Tactical Immigration Checkpoint (5:29)

Checkpoints are generally effective only against vehicular traffic because pedestrians tend to find ways around them. However, if strategically placed, it is possible for checkpoints to act as temporary deterrents against pedestrian intruders. (5)

The Need for Technological Innovation

With half a million people crossing the border illegally each year, it is obvious that the border patrol is not able to stop all of the illegal cross-border inflow of aliens, drugs, and other

contraband. It is felt that the U.S. Border Patrol is undermanned and underfunded but Washington has done little to change this situation; even after the events of September 11, 2001 (28). In May of 2006, President Bush announced a $1.9 billion plan which has placed nearly 6000 National Guard troops to the U.S. border with Mexico (29). Unfortunately, National Guard troops now stationed on the Mexican border cannot be fully utilized because

> under existing rules of force signed by the Department of Defense and border state governors, soldiers are not supposed to stop, arrest, or shoot armed illegal immigrants. They are instructed only to look, listen and report their location to the Border Patrol. (30)

While putting the National Guard on the border may be a great idea, ordering the Guard to maintain the status of observers turns them into nothing more than a human sensor network. This job can be done more effectively, and possibly cheaper, with an electronic sensor network.

As with almost all organizations, the largest part of the Border Patrol's budget goes to payroll. This makes it very difficult to add additional manpower because it requires a large budgetary increase. In fact, even if the Border Patrol was appropriated enough funds to double its manpower, it would not guarantee significantly better results. After all, the Border Patrol is currently only able to capture an estimated 25% of intruders (12). Even if the Border Patrol managed to cut down the rate of illegal border crossings to half, or even one-fourth, of their current rate, there would still be a serious illegal immigration problem.

This is where technological innovation can be used as a force enhancer. Installing smart fences on the nation's borders would allow agents to know exactly when and where a breach occurs. Installing sensor packages, including radar, video cameras, infrared cameras, seismic sensors, and other advanced technologies, would allow agents to detect and track intruders in real time. This would eliminate the most time consuming part of an agent's job (searching for

intruders) and allow the agent to focus most, if not all, of his or her time on apprehending intruders.

Thus, rather than just adding more agents, it is essential that the Border Patrol provide its agents with the latest advanced technology to help them do their jobs safely and much more effectively. In fact, properly-employed technology acts as a force multiplier for border security personnel. (13)

New Approach

While purchasing new sensors and other technologies for the border patrol is very important, the funds will be less effective if the new technology is not properly employed. Given a set budget, it is extremely important that the border patrol be able to balance the training and sustainment of personnel as well as technology and infrastructure (13). The Border Patrol must be able to identify how many sensors they need to buy and where they should place them, based on reasonable budgetary constraints. A computer software-focused approach will be employed to help the Border Patrol make this important decision.

It was decided to use a network manually created and overlaid onto a map using Google Inc.'s free Google Earth software. The node coordinates are imported into Excel and used to populate an optimization model. The model is created using various techniques to optimize the purchase and placement of electronic sensors under pre-determined budgetary constraints.

Data Development

The development of the network was done in several steps. First a location was picked for the network. Then, the nodes of the network were overlaid on a map of the network location. Finally, the coordinate locations of the nodes were extracted and imported into Excel.

Location.

A 20 kilometer section of the U.S.-Mexico border near Calexico, California was chosen as the location for the sensor network. Overhead satellite imagery provided by Google Earth suggests the area is comprised almost entirely of level farmland, with a uniform elevation and few obstacles. However, aside from the overhead satellite image, little else is known about the location. In lieu of a thorough on-ground inspection of the location, the network created from the image is treated as a notional network. The assumptions that have been made about this network may or may not represent the actual conditions at the location.

Google Earth.

The database for the network was created using Google Earth's "placemark" feature with a simple circle and diamond node representation.

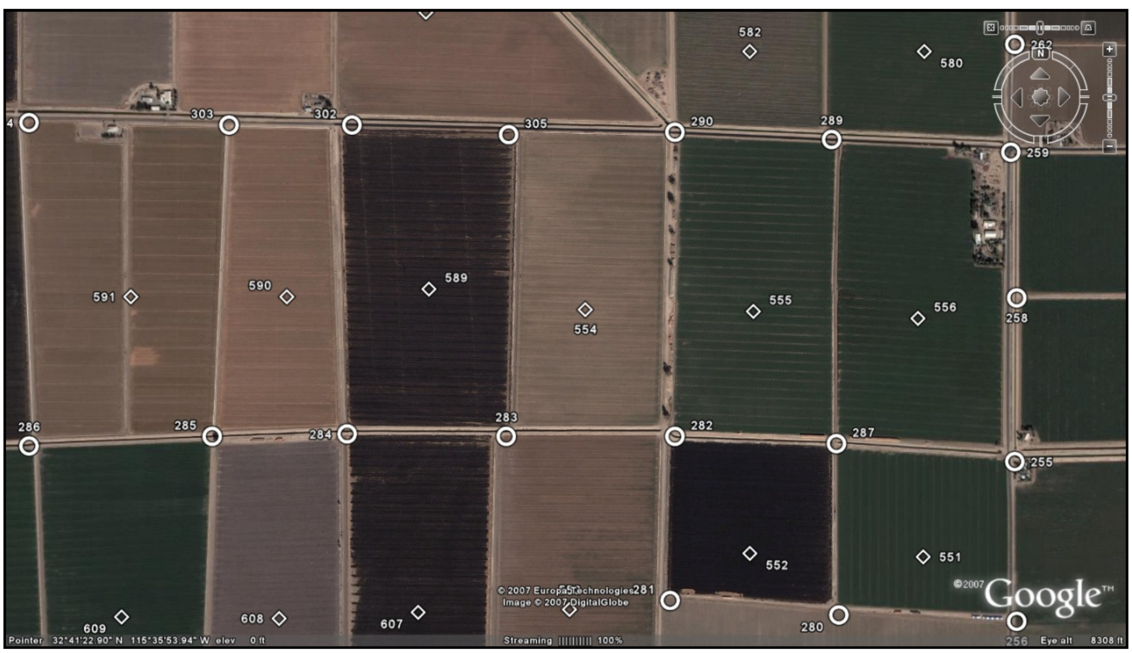


Figure 6. Intersection nodes and centers of land parcels (6)

Circles were used to indicate an intersection between two or more roads, while diamonds were used to represent centers of land parcels. Because the land near Calexico is mostly

agricultural, there was a need to differentiate between the two types of nodes (i.e. the nodes indicating centers of land masses is only used for detection and capture of walking intruders).

Data Conversion.

Google Earth saves user-created data points in a xml document called a kml file. Although kml files are text files, their format makes it difficult to directly import their contents into Excel. The problem arises from the fact that kml files are written in extensible markup language (xml) and contain a number of rows for each node in the network. These rows contain xml tag information as well as the node coordinates. The software program GEPath was used to parse the Google Earth kml file and create a simple spreadsheet grid with the number of each node, its longitude, and its latitude. (7)

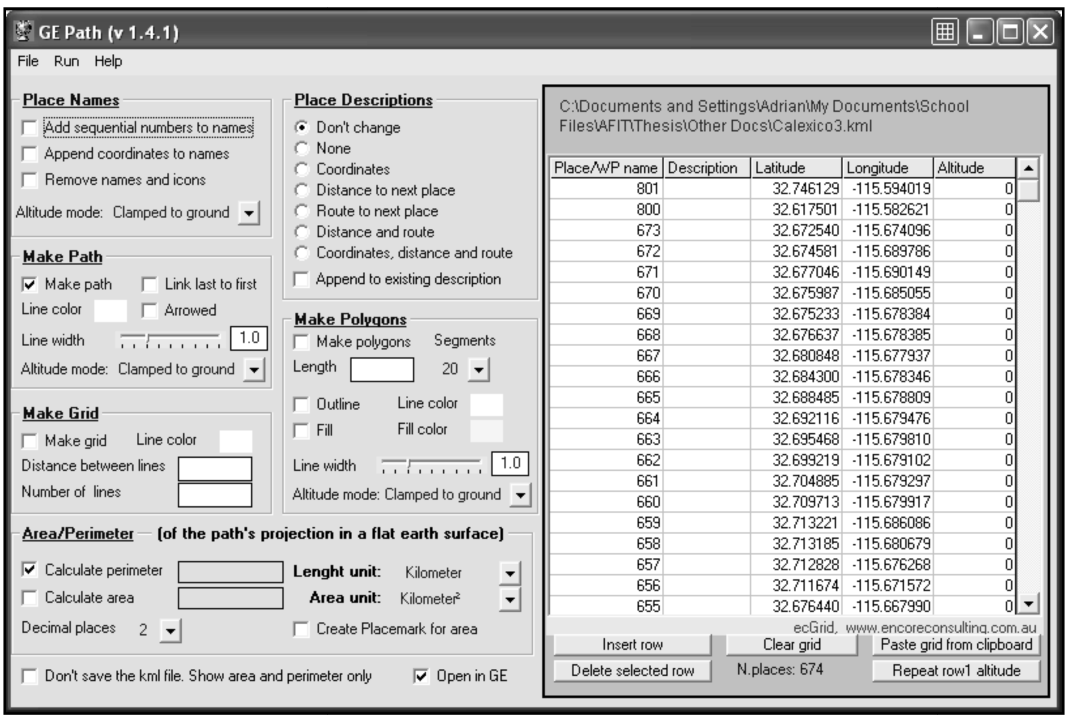


Figure 7. GEPath with data from the Calexico kml file (7)

Microsoft Excel Import.

The grid created by GEPath (Figure **7**), was copied into an Excel document. Once copied into Excel, it became the foundation for the sensor placement model.

	A	B	C	D	E
1	Place/WP name	Description	Latitude	Longitude	Altitude
2	1		32.651651	-115.684242	0
3	2		32.652055	-115.671424	0
4	3		32.652151	-115.669898	0
5	4		32.652492	-115.665565	0
6	5		32.652738	-115.662701	0
7	6		32.653394	-115.662622	0
8	7		32.653865	-115.65637	0
9	8		32.65424	-115.651822	0
10	9		32.654598	-115.647554	0
11	10		32.655333	-115.638505	0
12	11		32.655853	-115.632227	0
13	12		32.656295	-115.625752	0
14	13		32.656684	-115.62077	0
15	14		32.657038	-115.616463	0
16	15		32.65725	-115.611903	0

Figure 8. Data imported into Excel

Model Development

Using the data imported from Google Earth, an iterative sensor placement model was developed. The model maximizes the probability of detecting intruders by optimizing the build-up of a distributed sensor network subject to a budgetary constraint. Several different optimization algorithms are developed for use with the model. Additionally, the model is compatible with the commercial OptQuest solver software.

Variables.

For each of the 673 nodes in the Calexico network, there is a binary variable s_i, $i{:}1\text{-}673$, which is used to select the location of sensors on the network. s_i is equal to one at nodes with sensors and zero at nodes without sensors.

Distances between Nodes.

A distance matrix, δ_{ij}, was created calculating the distance between all pairs of nodes (673x673) in the network.

	A	B	C	D	E	F	G	H	I	J
1	Node	Latitude	Longitude	0.017453	Node	1	2	3	4	5
2	1	32.651651	-115.684242		1	0	1.20008	1.34316	1.749896	2.018965
3	2	32.652055	-115.671424		2	1.20008	0	0.143169	0.550307	0.819632
4	3	32.652151	-115.669898		3	1.34316	0.143169	0	0.407156	0.676492
5	4	32.652492	-115.665565		4	1.749896	0.550307	0.407156	0	0.269342
6	5	32.652738	-115.662701		5	2.018965	0.819632	0.676492	0.269342	0
7	6	32.653394	-115.662622		6	2.031984	0.836833	0.6946	0.293017	0.073268
8	7	32.653865	-115.65637		7	2.619232	1.422709	1.279901	0.873687	0.605407
9	8	32.65424	-115.651822		8	3.046762	1.849921	1.707006	1.30035	1.031404
10	9	32.654598	-115.647554		9	3.448033	2.251026	2.108059	1.701233	1.432104
11	10	32.655333	-115.638505		10	4.29855	3.101274	2.958246	2.551258	2.281992
12	11	32.655853	-115.632227		11	4.888722	3.691354	3.548306	3.141273	2.871977
13	12	32.656295	-115.625752		12	5.49642	4.298817	4.155741	3.748658	3.479335
14	13	32.656684	-115.62077		13	5.964507	4.766842	4.623758	4.21666	3.94733
15	14	32.657038	-115.616463		14	6.369356	5.171674	5.028585	4.62148	4.352147
16	15	32.65725	-115.611903		15	6.796312	5.598398	5.455291	5.048161	4.778821
17	16	32.658381	-115.599025		16	8.00754	6.809708	6.666602	6.259472	5.990132
18	17	32.659262	-115.585936		17	9.235853	8.037834	7.894713	7.487568	7.218227
19	18	32.659525	-115.581581		18	9.64424	8.446146	8.303021	7.895872	7.626531

Figure 9. Distance matrix

The distances were calculated using the Great Circle Distance formula. (31; 32)

$$\delta_{ij} = 60 * \operatorname{acos}\Big(\sin(\alpha * \omega) * \sin(\beta * \omega) + \cos(\alpha * \omega) * \cos(\beta * \omega) * \cos\big((\mu - \nu) * \omega\big)\Big) * \left(\frac{1}{\omega}\right) * \psi$$

x = arc length

α = Latitude of Node i β = Longitude of Node i

μ = Latitude of Node j ν = Longitude of Node j

$\omega = \frac{\pi}{180}$ ψ = 1.8520 (kilometers per nautical mile)

Note that, while the Calexico network may not require the use of the Great Circle Distance formula (rectilinear calculations could have been used due to the relatively short distances involved), the Great Circle Distance formula was used in order to provide scalability to

the model. In addition, the complete grid of 673x673 distances is calculated only one time and the model is not encumbered by distance calculations.

Sensor Locations.

The model assumes sensors are placed together in packages. Each node has a defined package which may contain all or some of the sensors. The binary input parameters s_{ik}, *i:1-673, k:1-4* describe which sensor types (*k*) can be placed at each node. i.e. $s_{ik}=1$ if the node *i* can host sensor type *k* and $s_{ik}=0$ if the node *i* cannot host sensor type *k*. These inputs to the model are made based on geographical, political, and economic considerations and vary based on the location of the network. For the network tested in this research, it was decided not to place any seismic sensors at intersection nodes because their effectiveness to detect intruders on foot will likely be compromised by legal vehicular traffic. Once the packages are determined, a sensor selection (s_i) at a node selects all sensor types available to that node. This assumption creates fewer physical sensor locations making it easier to secure and cheaper to maintain the network than if each sensor is allowed its own location. In addition, by placing sensors in packages, the number of variables (s_i) is limited to 673 regardless of the number of sensor types being used.

Sensor Ranges and Probability Distributions.

A review of sensor placement literature has revealed a couple of different methods used to define a sensor's probability of detection. The first, an unbounded method, uses a parameter α to obtain a probability of detection of an intruder by a sensor which varies exponentially with the distance, δ, between the intruder and the sensor. Using this method, the probability of detection becomes $e^{-\alpha\delta}$ (24:1610). The second method is bounded, but assumes a binary probability of detection (*d*) so that $d=1$ when the intruder is within range of the sensor and $d=0$ when the intruder is out of range of the sensor. (25:3)

For the current model, a new bounded method for determining probability of detection is developed. This method places lower and upper bounds on each sensor's probability of detection and assumes a continuous distribution between those bounds. The cumulative distribution function (cdf) of the Beta probability distribution is used to model the detection probability curve for each sensor within its prescribed range. The Beta probability distribution was chosen for its extreme versatility. By changing the distribution's shape parameters (α and β), the beta density function can be decreasing, increasing, convex, concave, uniform, and so forth. For a more detailed description of how the parameters α and β affect the shape of the Beta probability distribution, see Appendix A. The flexibility provided by the Beta distribution allows the user to change the parameters, and therefore the curve of the probability distribution, to match that of the sensors being used.

The Beta distribution is available as a function in Excel and requires 5 input parameters: δ = distance to be evaluated, α and β are shape parameters, and a and b are the lower and upper bounds. The Beta cdf is equal to 0 at the lower bound and 1 at the upper bound. The Beta cdf will be assumed to indicate the probability of a miss (m) with 0% chance of a miss at the lower bound (set to zero) and 100% chance of a miss at the upper bound or beyond (set to the range of each sensor type). Note that, while this is a reasonable assumption, it is notional and has not been validated from actual sensor data. It is assumed that sensors have a 100% chance of detection at a distance of 0 kilometers, and a 0% chance of detection at a given distance, with decreasing probability of detection between the given bounds. In order to obtain a probability of detection (d) equal to 1 at the lower bound (shortest distance) and 0 at the upper bound (longest distance), $d=1-m$ is used.

Probability of Missing (1- Probability of Detection).

For every set of nodes i and j in the sensor field, and for every sensor type k, the probability m_{ijk}, which denotes the probability that a target at node j is missed by a sensor of type k at node i, is calculated. Conversely, the value d_{ijk} indicates the probability of detection and can be (but is not) calculated by using $d_{ijk} = 1 - m_{ijk}$. Also, given a specific set of nodes i and j in the network, the probabilities m_{ijk} and m_{jik} are assumed symmetric because the network used in this research consists almost entirely of level farmland, with a uniform elevation and few obstacles. However, the probabilities m_{ijk} and m_{jik} can differ in the presence of obstacles and elevation differences. In order to account for these differences, the input parameter e_{ij}, $i,j{:}1\text{-}673$ can be used. For networks with varying elevations and other obstacles, the values in the e_{ij} matrix can be set anywhere between 0 and 1 allowing the probability of detection at individual sets of i-j nodes to be at its greatest value ($e_{ij}=1$), its lowest value ($e_{ij}=0$), or to be degraded ($0<e_{ij}<1$). The level of degradation for a particular i-j arc should be based on the observed degradation created by the elevation change or obstacle in question. The formula for the detection probability is:

$$d_{ijk} = 1 - m_{ijk} = (1 - \; Beta(\delta_{ij}, \alpha_k, \beta_k, a_k, b_k))e_{ij}; for\ i,j{:}\, 1 - 673, k{:}\, 1 - 4$$

But the above formula is not used in the model. Instead, the above formula is used to compute the formula for miss probability, m_{ijk}, which is then used in the model.

$$m_{ijk} = 1 - e_{ij} + Beta(\delta_{ij}, \alpha_k, \beta_k, a_k, b_k)e_{ij}; for\ i,j{:}\, 1 - 673, k{:}\, 1 - 4$$

Furthermore, since this research assumes level ground and no obstacles, all values $e_{ij}=1$, $i,j{:}1\text{-}673$ for the purposes of this research. With all $e_{ij}=1$,

$$m_{ijk} = Beta(\delta_{ij}, \alpha_k, \beta_k, a_k, b_k); for\ i,j{:}\, 1 - 673, k{:}\, 1 - 4$$

For combining probabilities, it is assumed that sensor detections are independent, i.e. if a sensor detects an intruder at a node with probability a, and another sensor detects the intruder

with a probability b, the combined probability of detection = *1- (1-prob. of detect. a)(1-prob. of detect. b)*. Furthermore, it is assumed that some sensor types are capable of both daytime and nighttime operation, while others are capable of only daytime or nighttime operation. For each node in the network, probabilities of detection for daytime and nighttime, d_i^d and d_i^n respectively, are computed as follows:

$$d_i^d = 1 - \prod_{j,k} m_{ijk}; for\ i, j: 1 - 673\ , k: day$$

and

$$d_i^n = 1 - \prod_{j,k} m_{ijk}; for\ i, j: 1 - 673\ , k: night$$

The daily probability of detection, p_i, *i:1-673* is defined as the average between the daytime and nighttime probabilities of detection.

$$p_i = average\left(d_i^d, d_i^n\right); for\ i: 1 - 673$$

Average daily probability of detection, *AvgCov*, and minimum daily probability of detection, *MinCov*, are also defined as follows:

$$Average\ Coverage = AvgCov = \ average(p_i)\ ; for\ i: 1 - 673$$

$$Minimum\ Coverage = MinCov = \ minimum(p_i)\ ; for\ i: 1 - 673$$

Note that the d_i^d and d_i^n calculations are made under the assumption of independent probabilities. Independence is a notional assumption used in this and other sensor placement literature (21; 24; 25). In actual practice, a certain amount of correlation may exist between sensors (25:4). However, if the amount of correlation is determined to be statistically insignificant, the independent assumption can continue to be used. Otherwise, the formulas may need to be modified to account for correlation. Also, this research does not address sensor fusion, i.e. the process by which the data from the various sensors is combined and processed.

Total Cost Calculation.

The total cost of building the proposed sensor network is calculated and used as a parameter forcing adherence to a budgetary constraint. By design, sensors are placed at as many nodes as possible in order to maximize coverage. Absent a budgetary constraint, sensors would be placed at every node in the network. There are five cost parameters. *InfCost* is the infrastructure cost for building at a node. *InfCost* is the same at every node with sensors present. The input parameters $SensCost_k$, $k{:}1\text{-}4$ define individual costs for purchasing and installing the four sensor types. *Total Cost*, is the total cost for building the network and can be calculated from the five cost parameters:

$$Total\ Cost = \sum_{i} s_i(\,InfCost + \sum_{k} s_{ik} * SensCost_k)\,, i{:}\,1 - 673, k{:}\,1 - 4$$

In the *Total Cost* calculation, s_i is the binary selection variable, equal to 1 at nodes with sensors and 0 at nodes without sensors, and s_{ik} is the binary input parameter describing which sensor types are placed at a node if $s_i=1$ at that node. The total cost calculation is re-computed at each iteration.

Constraints

The first constraint requires the binary selection of nodes for sensor placement. i.e. s_i, $i{:}1\text{-}673$ is binary. A number of additional constraints are used to compel the solution to exhibit desired attributes.

Budget.

A notional *Budget* is assumed to be available for the build-up of the network and a constraint is created so that the total cost cannot exceed the total budget ($Total\ Cost \leq Budget$).

Node Availability.

A binary input parameter v_i, i:1-673 is used to allow nodes to be turned on or off for sensor placement. The constraint $s_i \leq v_i$ turns nodes off if $v_i=0$. The parameter v_i is set to 1 (on) by default.

Node Preference.

An input parameter w_i, i:1-673 may be used to require certain pre-identified nodes to have a minimum coverage, p_i. (25:6) Note that meeting this constraint may require more assets than available under the budget and could result in an infeasible solution. For the notional network used to test the model developed in this research, there are no preferential nodes or zones (sets of nodes), but the model was designed to be able to use this constraint.

Solving the Model

Dhillon and Chakrabarty, describe two notional algorithms for sensor placement which provide "effective coverage and surveillance in distributed sensor networks" (24:1609). The algorithm called "MAX_AVG_COV" is an iterative algorithm which places one sensor at a time, without backtracking, until the average miss probability drops below a desired maximum, i.e. the average detection probability rises above a desired minimum. Similarly, the algorithm called "MAX_MIN_COV" is an iterative algorithm which places one sensor at a time, without backtracking, until the largest miss probability drops below a desired maximum, i.e. the smallest detection probability rises above a desired minimum.

The model developed in this research is solved using algorithms similar to the ones described by Dhillon and Chakrabarty. Additionally, the Premium Solver Platform for Excel, together with the OptQuest field-installable engine, is used to solve the model.

VBA (Visual Basic for Applications)

Two algorithms, *VBA-AvgCov* and *VBA-MinCov*, were developed using the Visual Basic for Applications (VBA) programming language in Excel. *VBA-AvgCov* is an iterative greedy algorithm which places one sensor package at a time on the network (selects one node at a time) until all funds are exhausted, i.e. when the *Total Cost* exceeds the *Budget*, the algorithm stops and returns the previous (last feasible) solution. At each iteration, the algorithm places a sensor package at the node that will affect the greatest incremental increase in the average coverage, *AvgCov*. There is no backtracking in this algorithm. *VBA-MinCov* works in a similar fashion but always chooses the node that will produce the greatest incremental increase in the minimum, as opposed to the average, coverage *MinCov*. However, since the first few iterations are likely to produce *MinCov=0*, because there are not yet enough sensors to cover all nodes, the algorithm chooses a sensor location which maximizes *AvgCov* until it reaches a point where *MinCov>0*. Then, for the first iteration where *MinCov>0*, and each subsequent iteration, the algorithm switches to choosing a sensor location at the node which maximizes *MinCov*. This algorithm also works without backtracking. The VBA code for both algorithms can be found in Appendix B.

OptQuest Solver.

The two algorithms described above are used as a baseline for the model However, in an effort to ensure good results, an attempt is made to improve upon the baseline solutions using commercial solver software. Unfortunately, due to its size and complexity, the model cannot be solved using the built-in Excel Solver. In fact, the model exceeds the number of variables and constraints that the Excel Solver can handle. It is also non-linear and non-smooth (due mostly to the use of 'min' and 'if' functions) and the Excel Solver requires linearity.

However, OptQuest solver engine from Frontline Systems, the makers of the Excel Solver, “employs metaheuristics such as tabu search and scatter search to solve nonsmooth optimization problems of up to 5,000 variables and 1,000 constraints.” (27)

> In the tabu search category of meta-heuristics, the essential idea is to 'forbid' search moves to points already visited in the (usually discrete) search space, at least for the upcoming few steps. That is, one can temporarily accept new inferior solutions, in order to avoid paths already investigated. This approach can lead to exploring new regions of *D* [the search space], with the goal of finding a solution by 'globalized' search. Tabu search has traditionally been applied to combinatorial optimization (e.g., scheduling, routing, traveling salesman) problems. (33)

> Scatter search operates on a set of solutions, the *reference set*, by combining these solutions to create new ones. The main mechanism for combining solutions is such that a new solution is created from the linear combination of two other solutions. The reference set may evolve [over time]. (34)

In order to use the OptQuest solver engine, two software packages need to be installed: the first is Frontline Systems’ Premium Solver Platform (PSP), and the second is the OptQuest solver engine itself. The combined package is simply referred to as the “OptQuest solver.” While the OptQuest solver can only guarantee optimality by complete enumeration, Frontline Systems claims that the OptQuest solver finds “remarkably good solutions with unprecedented speed” (27). In Chapter 4, this claim is evaluated by comparing the solver results against the results of the VBA algorithms.

Review

The models discussed above can be summarized as follows:

$Max\ AvgCov$	*(Maximize Average Coverage)*
s.t.	***subject to,***
$Total\ Cost\ \leq Budget$	*(budgetary constraint)*
$s_i \leq v_i$	*(node on/off constraint)*
$w_i \leq p_i$	*(minimum coverage constraint)*

And

$Max\ MinCov$	*(Maximize Minimum Coverage)*
s.t.	***subject to,***
$Total\ Cost\ \leq Budget$	*(budgetary constraint)*
$s_i \leq v_i$	*(node on/off constraint)*
$w_i \leq p_i$	*(minimum coverage constraint)*

There is one model with two objectives. The VBA algorithms developed above, and the OptQuest Solver, are used to solve the optimization model for each objective. *AvgCov* represents the average daily coverage over the entire network and *MinCov* represents the node in the network which has the least sensor coverage. The *Total Cost* is calculated by adding up infrastructure and sensor costs for all selected nodes in the network. The *Budget* is an input parameter. For nodes $i = 1\text{-}673$, s_i is a binary variable indicating selected nodes, v_i is a binary input parameter which, when set equal to 0, prevents node selection, p_i is the calculated value for daily coverage (as defined previously), and w_i is an input parameter used to force minimum daily coverage.

IV. Results and Conclusions

In Chapter 3, a model is developed for placement of distributed sensors on a network with the goals of maximizing minimum coverage and average coverage on the network. In Chapter 4, the model is tested using the methods proposed in Chapter 3. First, the VBA algorithms are tested. Then, the OptQuest solver is tested and the results are compared to the previous results. The computer used for the tests is an Asus A8jp laptop with an Intel Core 2 Duo 7200 processor and 2GB of RAM running Microsoft Excel 2007 and OptQuest solver 7.0. Run times are preserved for each of the runs but the computer must be used for other work at the same time as the Excel runs. Therefore, the quoted times may not represent the full capability of the computer being used.

Inputs

A number of inputs are required to run the model. The 673 node network developed in Chapter 3 is the primary data input to the model. Figure 10 shows a representation of this network in Google Earth with circles representing intersection nodes and diamonds representing centers of land parcels.

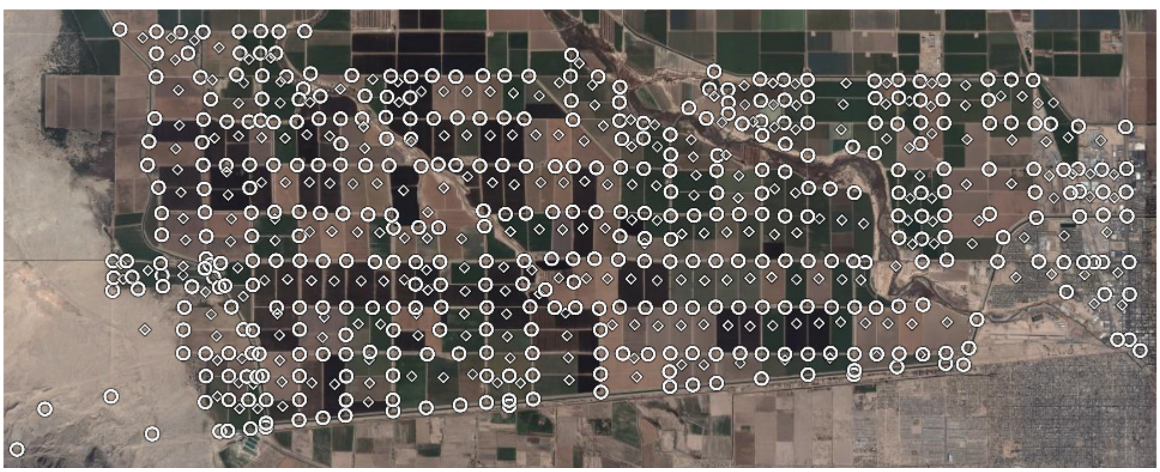


Figure 10. Complete Network

However, the full network, shown in Figure 10, contains a total of 47 nodes which are not included in the optimization for one of two reasons. A set of 4 nodes (orange nodes in Figure 11) are excluded because they are located away from the rest of the network, and a set of 43 nodes (red nodes in Figure 11) are excluded because they are located in the city. It is assumed that sensors will not be effective inside a city. For the *i* values corresponding to the eliminated nodes, the parameter v_i is set to 0 ($vi = 0$). The remaining 626 nodes have $v_i = 1$.

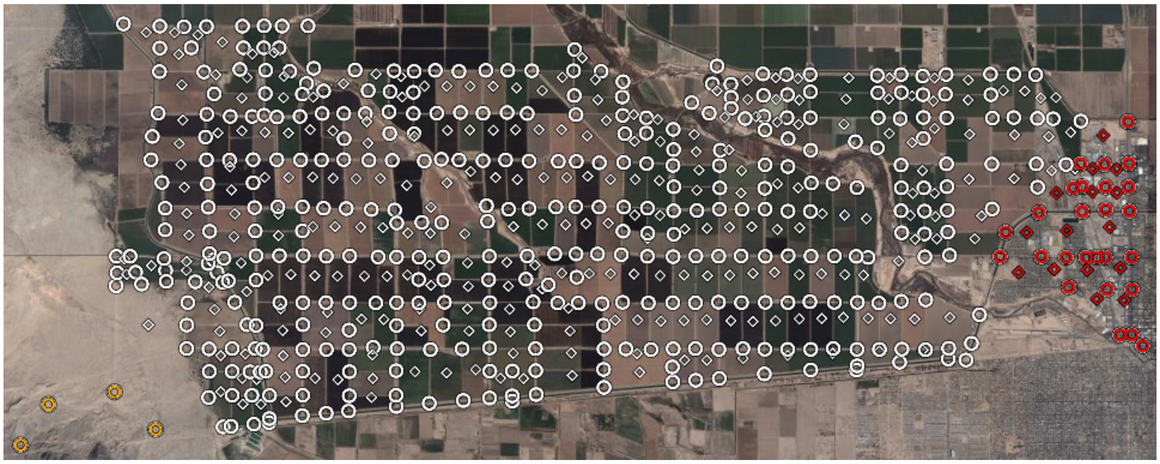


Figure 11. Network with eliminated nodes highlighted

Parameter Inputs.

The parameters chosen for these optimization runs (see Table 1 and Table 2) cannot be validated against true operational settings. Instead, the notional parameters selected appear “reasonable” for the purpose of these tests. For example, the 4 kilometer range assumption for the Ground Search Radar (sensor 3) is in line with the available data (3). The Budget parameter is varied in order to test the model under differing conditions.

Table 1. Sensor Input Parameters

Sensor	Sensor	Day/Night		Shape Parameters		Bounds		Cost
Name	k	day	night	α_k	β_k	a_k	b_k	$SensCost_k$
Regular Camera	1	yes	no	1.3	0.5	0	3	$5000
Infrared Camera	2	no	yes	0.8	1.2	0	1.2	$4000
Ground Search Radar	3	yes	yes	1.2	0.8	0	4	$6000
Seismic Sensor	4	yes	yes	0.6	1.3	0	0.8	$2000

Table 2. Additional Inputs

Total Construction Budget	*Budget*	$100,000; $300,000
Infrastructure Cost Per Node	*InfCost*	$7000

It is assumed that seismic sensors ($k=4$) cannot function effectively when placed at intersection nodes. For this reason, a seismic sensor will not be part of the sensor package if an intersection node is selected. Table 3 shows that the sensor types 1, 2, and 3 (s_{i1}, s_{i2}, and s_{i3}) will be placed at intersection nodes (i: $i=1\text{-}399$) while all 4 sensor types will be placed at center of land parcel nodes (i: $i=400\text{-}673$).

Table 3. Available Sensors

Node Type	i	s_{i1}	s_{i2}	s_{i3}	s_{i4}
Intersection Node	*1-399*	1	1	1	0
Center of Land Parcel	*400-673*	1	1	1	1

Lastly, due to the problem assumptions, as explained in Chapter 3, the parameters e_{ij} and w_i where i,j:$1\text{-}673$, were set to one and zero respectively.

Detection Curves.

The shape and bound parameters from Table 1 produce the following detection curves for the four sensor types:

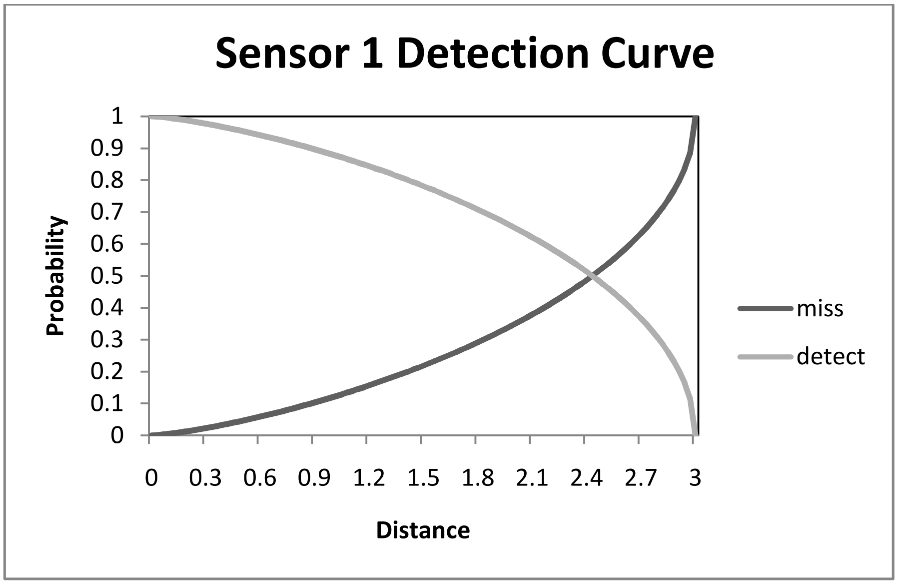


Figure 12. Sensor 1 Detection Probability Curve

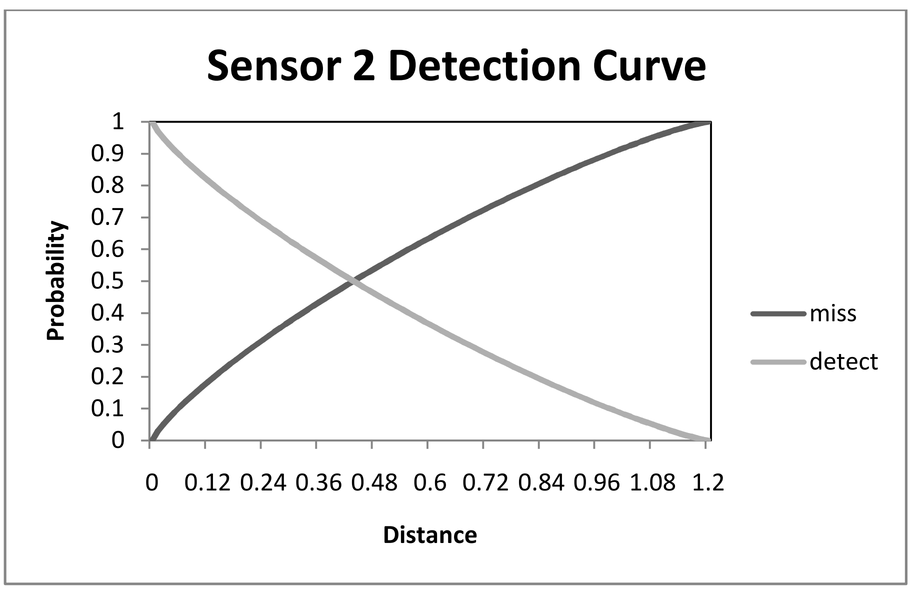


Figure 13. Sensor 2 Detection Probability Curve

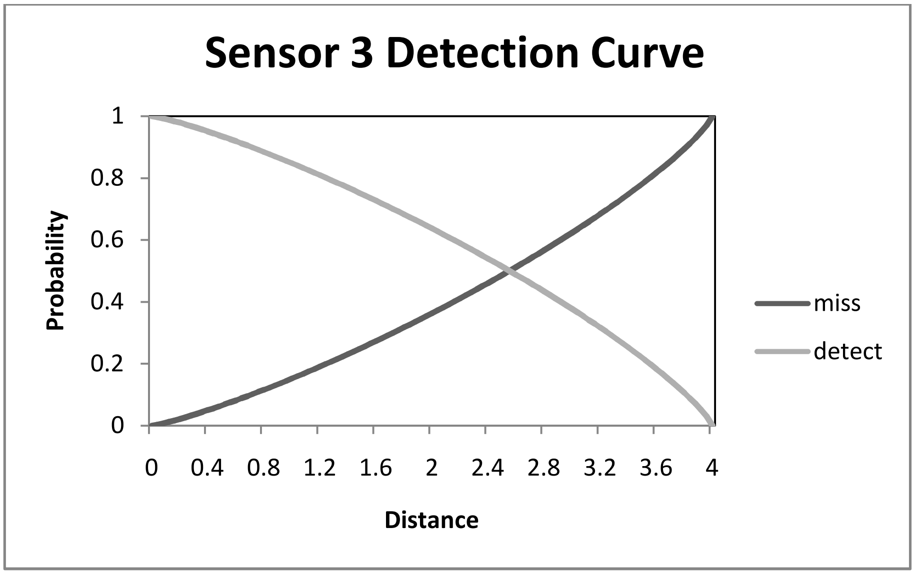


Figure 14. Sensor 3 Detection Probability Curve

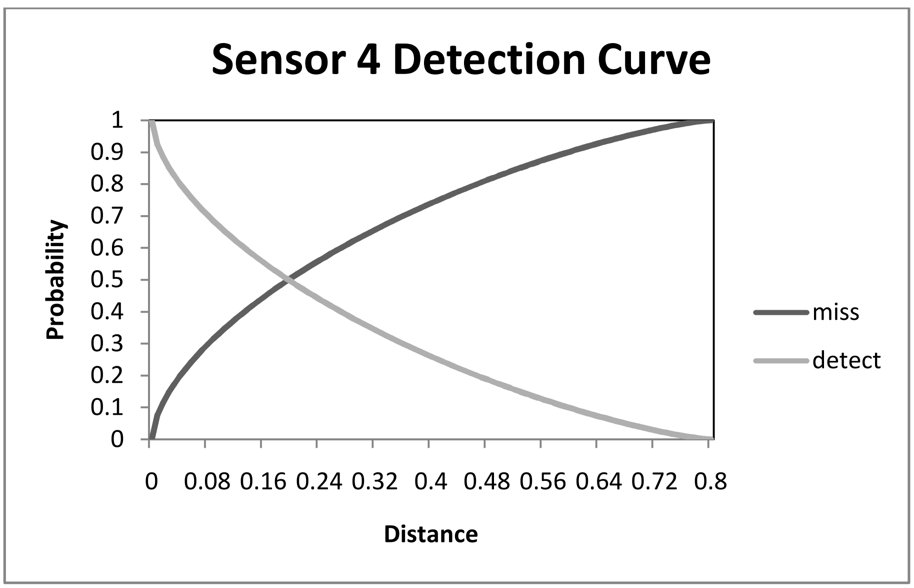


Figure 15. Sensor 4 Detection Probability Curve

Figures 12 through 15 show the detection and miss curves for each of the four sensor types, over their individual ranges. The curves are used to calculate point to point miss probabilities for each pair of nodes in the network and for each sensor type.

Results

Every test is completed two times; once assuming a budget of \$100,000 (*Run 1*), and a second time assuming a budget of \$300,000 (*Run 2*). The results of each of these two runs is shown below:

***VBA-AvgCov* Algorithm – *Run 1*.**

Table 4. *VBA-AvgCov* Algorithm *Run 1* Summary

Total Construction Budget	*Budget*	\$100,000
Total Construction Cost	*Total Cost*	\$96,000
Number of Sensors Selected	$\sum s_i$	4
Objective (Maximization)	*Average*(p_i)	0.7200
Run Time	*Hours*	3 (approx.)

Table 5. *VBA-AvgCov* Algorithm *Run 1* Probability Summary

	Sensor	**Day/Night**	**Variable**	**Average**	**(Max)/Min**	**Std. Dev.**
Probability of Missing	1	Day	m_{i1}	0.4268	(1.0000)	0.3280
	2	Night	m_{i2}	0.9547	(1.0000)	0.1378
	3	Both	m_{i3}	0.3420	(1.0000)	0.2410
	4	Both	m_{i4}	0.9826	(1.0000)	0.0935
Probability of Detection	1,2,4	Day	d_{id}	0.7795	0.0000	0.2668
	2,3,4	Night	d_{in}	0.6605	0.0000	0.2439
	1,2,3,4	Average	p_i	**0.7200**	0.0000	0.2525

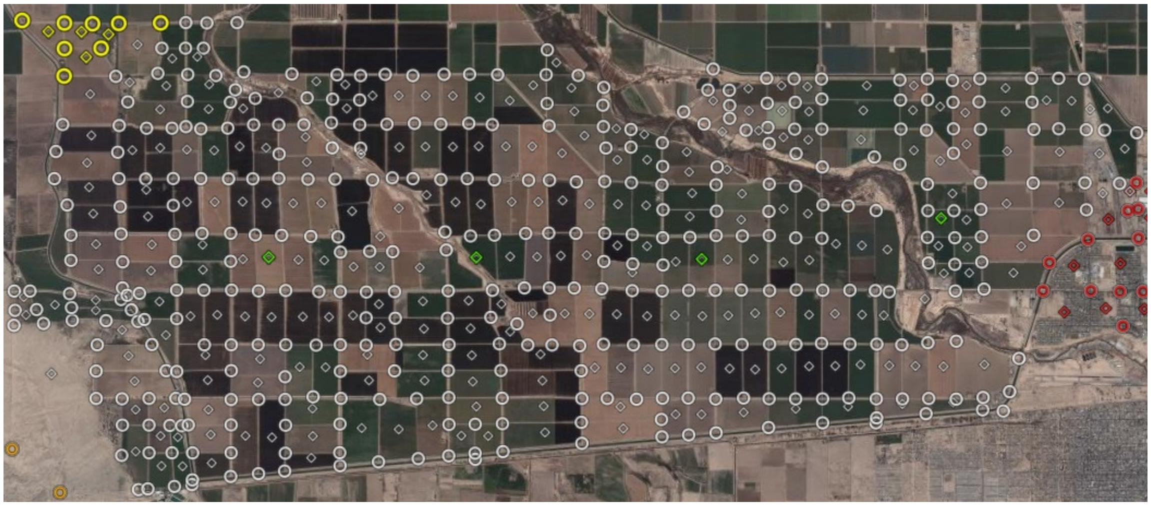


Figure 16. *VBA-AvgCov* Algorithm *Run 1* Visual Summary

Figure 16 shows the 4 nodes selected in *Run 1* highlighted in green. The selected sensors are placed in a relatively flat line across the center of the network. This selection produces an average coverage across the network of 72%. However, there are 12 nodes, highlighted in yellow, which are not covered by any of the sensors. Due to their location, away from the border and at the very back of the coverage area, these uncovered nodes, while not ideal, are not a great cause for concern.

***VBA-AvgCov* Algorithm – *Run 2*.**

Table 6. *VBA-AvgCov* Algorithm *Run 2* Summary

Total Construction Budget	*Budget*	\$300,000
Total Construction Cost	*Total Cost*	\$286,000
Number of Sensors Selected	$\sum s_i$	12
Objective (Maximization)	*Average*(p_i)	0.9700
Run Time	*Hours*	8 (approx.)

Table 7. *VBA-AvgCov* Algorithm *Run 2* Probability Summary

	Sensor	Day/Night	Variable	Average	(Max)/Min	Std. Dev.
Probability of Missing	1	Day	m_{i1}	0.0870	(1.0000)	0.1286
	2	Night	m_{i2}	0.8721	(1.0000)	0.2117
	3	Both	m_{i3}	0.0505	(0.3980)	0.0598
	4	Both	m_{i4}	0.9519	(1.0000)	0.1537
Probability of Detection	1,2,4	Day	d_{id}	0.9888	0.7183	0.0316
	2,3,4	Night	d_{in}	0.9513	0.6020	0.0603
	1,2,3,4	Average	p_i	**0.9700**	0.6643	0.0445

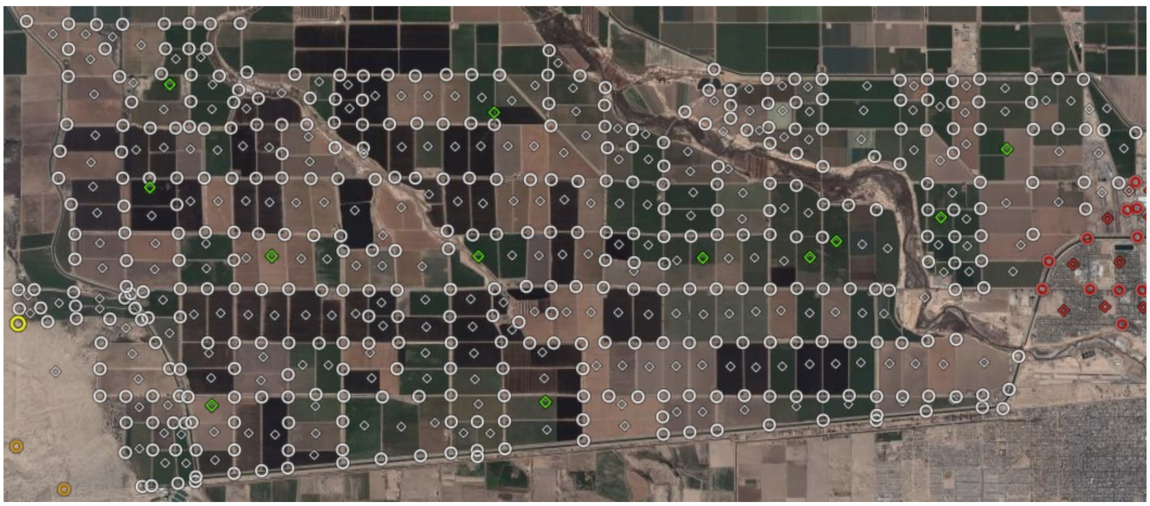


Figure 17. *VBA-AvgCov* Algorithm *Run 2* Visual Summary

Since the *VBA-AvgCov* algorithm does not backtrack, both runs are identical until the fourth sensor. However, while *Run 1* ended after placing the fourth sensor due to the smaller budgetary constraint, in *Run 2* the algorithm was allowed to continue placing sensors until the larger budget was exhausted. The *Run 2* solution improves the average coverage by 25% over the *Run 1* solution. Also, there are no longer any nodes with zero coverage. The node shown in yellow in Figure 17 is the node with the lowest average coverage at 0.664.

***VBA-MinCov* Algorithm – *Run 1*.**

The *VBA-MinCov* algorithm works the same way as the *VBA-AvgCov* algorithm until there is a minimum probability of detection greater than zero; at which point the algorithm starts to choose sensor locations in order to maximize the minimum probability of detection at each iteration. At the $100,000 budget level this threshold is not met, so *Run 1* for the *VBA-MinCov* algorithm is identical to *Run 1* for the *VBA-AvgCov* algorithm. *Run 2*, however, does produce results that are significantly different from those produced by *Run 2* of the *VBA-AvgCov* algorithm.

***VBA-MinCov* Algorithm – *Run 2*.**

Table 8. *VBA-MinCov* Algorithm *Run 2* Summary

Total Construction Budget	*Budget*	$300,000
Total Construction Cost	*Total Cost*	$294,000
Number of Sensors Selected	$\sum s_i$	13
Objective (Maximization)	*Minimum*(p_i)	.5156
Run Time	*Hours*	8 (approx.)

Table 9. *VBA-MinCov* Algorithm *Run 2* Probability Summary

	Sensor	**Day/Night**	**Variable**	**Average**	**(Max)/Min**	**Std. Dev.**
Probability of Missing	1	Day	m_{i1}	0.1303	(1.0000)	0.1905
	2	Night	m_{i2}	0.8827	(1.0000)	0.2159
	3	Both	m_{i3}	0.0831	(0.5684)	0.1098
	4	Both	m_{i4}	0.9526	(1.0000)	0.1595
Probability of Detection	1,2,4	Day	d_{id}	0.9698	0.5179	0.0722
	2,3,4	Night	d_{in}	0.9183	0.4316	0.1104
	1,2,3,4	Average	p_i	0.9441	**0.5156**	0.0893

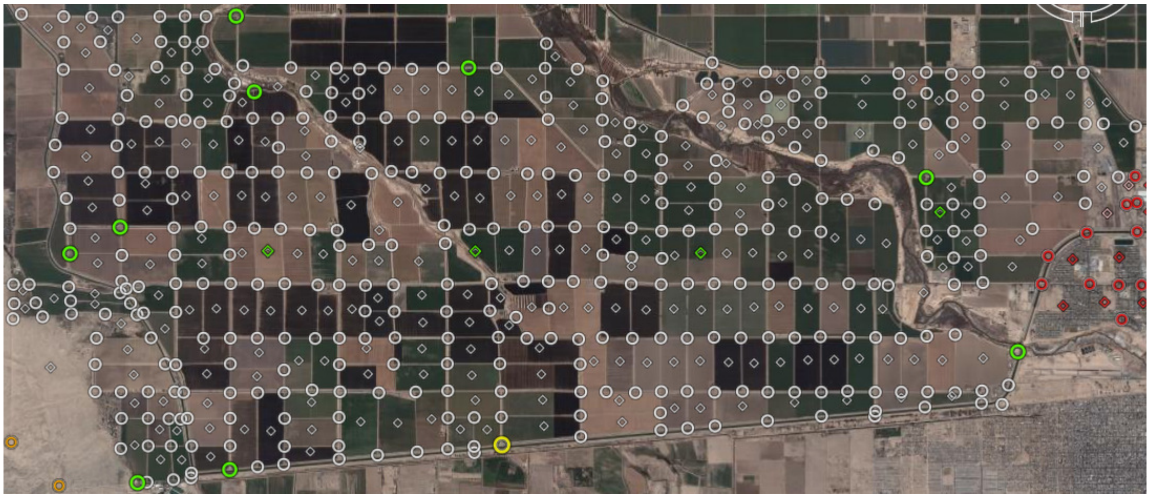


Figure 18. *VBA-MinCov* Algorithm *Run 2* Visual Summary

Table 9 shows the numerical, and Figure 18 shows the visual, result of the *VBA-MinCov* algorithm. Both numerically and visually, the results of the *VBA-MinCov* algorithm are inferior to those provided by the *VBA-AvgCov* algorithm. Using the *VBA-MinCov* algorithm, the average coverage is approximately 2.5% lower while minimum coverage is approximately 15% lower than the results gained from the *VBA-AvgCov* algorithm. The yellow node in Figure 18 is the node with the lowest average coverage (p_i) at 0.516. These results imply that the *VBA-AvgCov* algorithm is superior to the *VBA-MinCov* algorithm even when the objective is to maximize minimum coverage.

OptQuest Solver

The OptQuest solver (OQS) for Excel is used in an attempt to improve upon the results of the two VBA algorithms seen above. The OptQuest solver is given the same objectives and constraints as the VBA algorithms and two runs are completed for each objective just as before. All runs are completed using the OptQuest solver with 30,000 iterations unless mentioned otherwise. Appendix C shows all of the OptQuest solver inputs and options used for the following runs.

OQS-AvgCov – Run 1.

Table 10. *OQS-AvgCov Run 1* Summary

Total Construction Budget	*Budget*	\$100,000
Total Construction Cost	*Total Cost*	\$90,000
Number of Sensors Selected	$\sum s_i$	4
Objective (Maximization)	$Average(p_i)$	0.7403
Run Time	*Hours*	11:04:31

Table 11. *OQS-AvgCov Run 1* Probability Summary

	Sensor	Day/Night	Variable	Average	(Max)/Min	Std. Dev.
Probability of Missing	1	Day	m_{i1}	0.4064	(1.0000)	0.3049
	2	Night	m_{i2}	0.9511	(1.0000)	0.1421
	3	Both	m_{i3}	0.3303	(1.0000)	0.2053
	4	Both	m_{i4}	0.9813	(1.0000)	0.0951
Probability of Detection	1,2,4	Day	d_{id}	0.8077	0.0000	0.2327
	2,3,4	Night	d_{in}	0.6729	0.0000	0.2092
	1,2,3,4	Average	p_i	**0.7403**	0.0000	0.2175

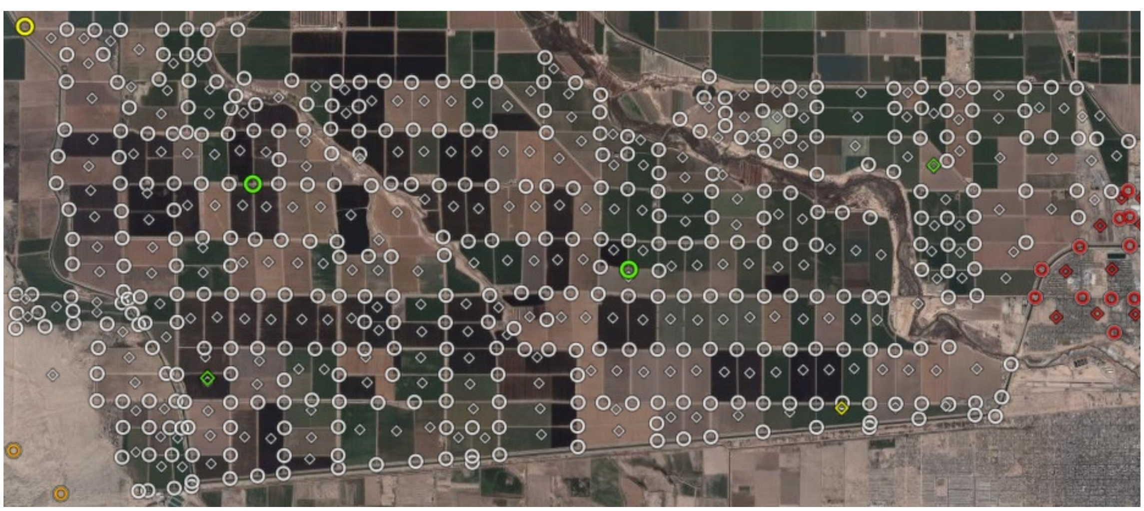


Figure 19. OQS-AvgCov *Run 1* Visual Summary

After 30,000 iterations, the OptQuest solver found a solution with 0.740 average coverage; approximately a 2% improvement over the *VBA-AvgCov Run 1* result. Also, while the

OptQuest solver did not find a solution with minimum coverage greater than zero, it did reduce the number of uncovered nodes (yellow nodes in Figure 19) to 2, down from 12 under the *VBA-AvgCov Run 1* result. The standard deviation is also lower using the OptQuest solver.

OQS-AvgCov – Run 2.

Table 12. *OQS-AvgCov Run 2* Summary

Total Construction Budget	*Budget*	$300,000
Total Construction Cost	*Total Cost*	$298,000
Number of Sensors Selected	$\sum s_i$	13
Objective (Maximization)	*Average*(p_i)	0.9795
Run Time	*Hours*	9:58:32

Table 13. *OQS-AvgCov Run 2* Probability Summary

	Sensor	Day/Night	Variable	Average	(Max)/Min	Std. Dev.
Probability of Missing	1	Day	m_{i1}	0.0574	(0.4269)	0.0618
	2	Night	m_{i2}	0.8436	(1.0000)	0.2269
	3	Both	m_{i3}	0.0387	(0.3264)	0.0390
	4	Both	m_{i4}	0.9376	(1.0000)	0.1675
Probability of Detection	1,2,4	Day	d_{id}	0.9957	0.9012	0.0095
	2,3,4	Night	d_{in}	0.9638	0.6736	0.0395
	1,2,3,4	Average	p_i	**0.9797**	0.7874	0.0241

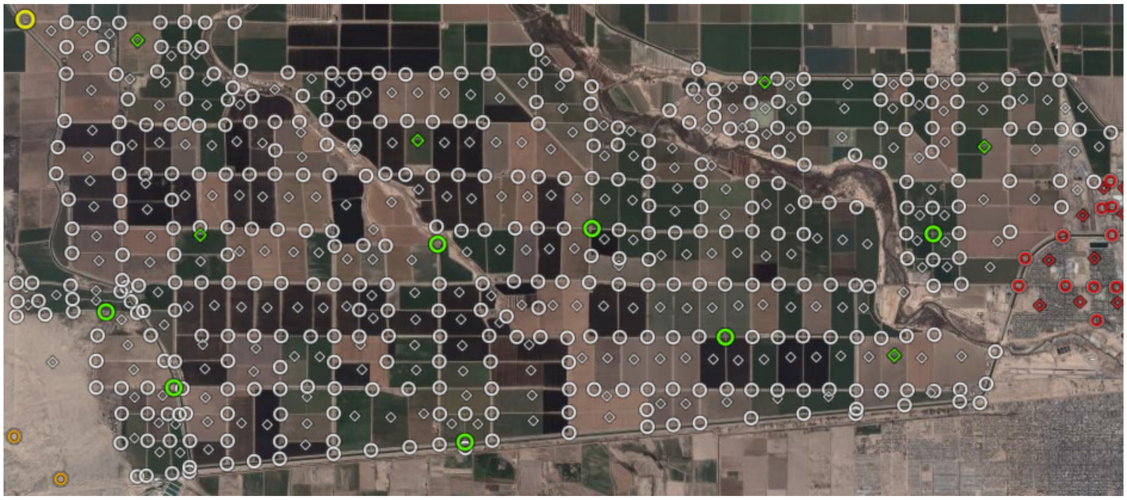


Figure 20. *OQS-AvgCov Run 2* Visual Summary

The OptQuest solver produced a solution with a 1.5% increase in average coverage, and 12% increase in minimum coverage, over the *VBA-AvgCov* algorithm *Run 2* solution. The yellow node in Figure 20 is the node with the lowest average coverage (p_i) at 0.787.

OQS-MinCov – Run 1.

After 30,000 iterations, the OptQuest solver was not able to find a solution with minimum coverage greater than zero (given the $100,000 budgetary constraint for *Run 1*). The OptQuest solver can only search for one objective at a time and, failing to find a solution with *MinCov* > 0, it simply returned an all zero solution set, i.e. no nodes were selected for sensor placement. While this failure cannot guarantee that a solution with *MinCov* > 0 does not exist, it is part of a larger pattern which appears to point in that direction. However, the OptQuest solver can only prove *MinCov* = 0 by complete enumeration. Therefore, additional runs may be able to find a solution with *MinCov* > 0.

OQS-MinCov – Run 2.

Table 14. *OQS-MinCov Run 2* Summary

Total Construction Budget	*Budget*	\$300,000
Total Construction Cost	*Total Cost*	\$294,000
Number of Sensors Selected	$\sum s_i$	13
Objective (Maximization)	$Minimum(p_i)$	0.8244
Run Time	*Hours*	10:53:11

Table 15. *OQS-MinCov Run 2* Probability Summary

	Sensor	Day/Night	Variable	Average	(Max)/Min	Std. Dev.
Probability of Missing	1	Day	m_{i1}	0.0679	(0.5777)	0.0750
	2	Night	m_{i2}	0.8417	(1.0000)	0.2290
	3	Both	m_{i3}	0.0437	(0.2404)	0.0397
	4	Both	m_{i4}	0.9368	(1.0000)	0.1668
Probability of Detection	1,2,4	Day	d_{id}	0.9944	0.8892	0.0116
	2,3,4	Night	d_{in}	0.9590	0.7596	0.0406
	1,2,3,4	Average	p_i	0.9767	**0.8244**	0.0255

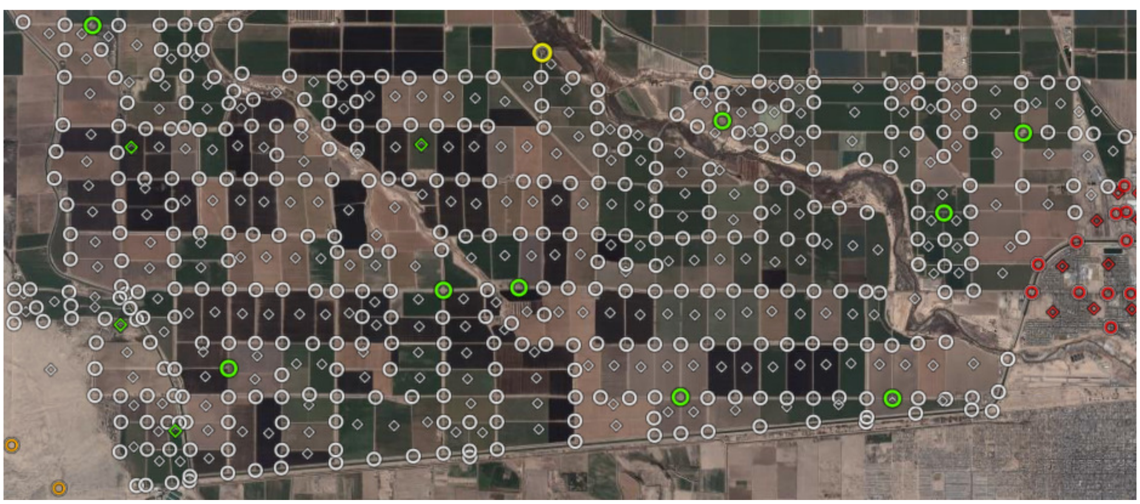


Figure 21. *OQS-MinCov Run 2* Visual Summary

The *OQS-MinCov Run 2* solution has a 3.3% higher average coverage and a 30.9% higher minimum coverage than the *VBA-MinCov* algorithm Run 2 solution. Also, the *OQS*-

MinCov Run 2 solution has an average coverage almost identical to the *OQS-AvgCov Run 2* solution (0.3% smaller) and a 3.7% higher minimum coverage than the *OQS-AvgCov Run 2* solution. The yellow node in Figure 21 is the node with the lowest average coverage (p_i) at 0.824.

Summary

The OptQuest solver produced consistently better results than the VBA algorithms. Table 16 shows that, under the Max *AvgCov* objective, the OptQuest solver produced the highest average coverage and, under the Max *MinCov* objective, the OptQuest solver produced the highest minimum coverage. The VBA algorithms produced worse results in both cathegories.

Table 16. Result Summary

	Objective	Type	Average(p_i)	Minimum(p_i)	Std. Dev.(p_i)
Run 1	Max-Avg	VBA	**0.7200**	0.0000	0.2525
		Solver	**0.7403**	0.0000	0.2175
	Max-Min	VBA	**0.7200**	0.0000	0.2525
		Solver	0.0000	**0.0000**	0.0000
Run 2	Max-Avg	VBA	**0.9700**	0.6643	0.0445
		Solver	**0.9797**	0.7874	**0.0241**
	Max-Min	VBA	0.9441	**0.5156**	0.0893
		Solver	0.9767	**0.8244**	0.0255

Model Behavior Analysis

Feasibility.

Under the $100,000 budgetary assumption, the OptQuest solver cannot find a feasible solution for the maximization of the minimum coverage (*OQS-MinCov Run 1*). The budget allows only 4 sensor packages to be purchased and, with 4 sensor locations, the objective, Max *MinCov,* appears to equal zero, i.e. there is always at least one node which is not covered by the solution. Since it cannot find a feasible solution, the OptQuest solver returns an empty solution

set, i.e. no nodes are selected for sensor placement. Furthermore, although the *VBA-MinCov* algorithm does produce an answer under the $100,000 budgetary assumption (*Run 1*), the answer is based solely on the secondary objective (Max *AvgCov*) and is identical to the answer produced by *Run 1* of the *VBA-AvgCov* algorithm.

Node Separation.

Under the $100,000 budgetary assumption, sensors are approximately equidistant from each other, while under the $300,000 budgetary assumption, sensors are placed both close together and far apart. This leads to a visual solution, Figure 21, which appears to have large gaps in it. However, due to overlapping coverage created by additional sensors, the apparent gaps are actually covered very well; a fact which is attested to by a Standard Deviation value of approximately 2.5%. Figure 22 shows the coverage for sensor 4 (Ground-Search Radar) for the *OQS-MinCov Run 2* solution. The apparent gaps in Figure 21 are very well explained by the coverage map shown in Figure 22. The figure also explains the low standard deviation.

Figure 22. OQS-MinCov *Run 2* Sensor 4 Coverage Map

Figure 22 shows that the number of sensors covering a particular node is directly related to that node's distance from the closest sensor. Since a sensor's probability of detection drops as the distance from the sensor increases, the solution covers the nodes that are farther from a sensor with additional sensors.

Solution Speed.

The four OptQuest solver-based solutions shown above took 30,000 iterations, and as long as 11 hours, to obtain. In some instances, it may be necessary to obtain solutions faster than they can be obtained with 30,000 iterations. Four additional runs of the OptQuest solver with the Max *AvgCov* objective in order to test the solver's ability to find good solutions quickly. The additional runs were set to 500, 1000, 5000, and 10,000 iterations respectively. Table 17 shows the results of these runs as well as the original result for 30,000 iterations. A budgetary constraint of $300,000 was used for all runs.

Table 17. *OQS-AvgCov* Iteration Comparison

Iterations	*AvgCov*	*MinCov*	Std. Dev.	Run Time
500	**0.9648**	0.5870	0.0620	00:34:21
1000	**0.9735**	0.6737	0.0426	00:36:40
5000	**0.9709**	0.6314	0.0435	01:49:17
10000	**0.9787**	0.8026	0.0270	04:53:02
30000	**0.9797**	0.7874	0.0241	09:58:32

Table 17 shows that a good result can be achieved within as few as 1000 iterations. Also note that the result after 1000 iterations is better than the result after 5000 iterations. Since the OptQuest solver is allowed to pick different random number sequences with each run (see Appendix C, Figure 28) the results can and do differ with each run. However, the general trend shows that the results improve with the number of iterations. If time is not of the essence, it is suggested that at least 30,000 iterations be completed.

Conclusion

The research developed a "proof-of-concept" model for distributed sensor placement optimization for border security using Microsoft Excel and Frontline Systems' OptQuest solver. The road network of Calexico, California, which was used for this model, was manually created using Google Inc.'s Google Earth application and transferred to Excel using the free GEpath application. The model optimizes the placement of electronic sensors, in order to maximize the average per-arc probability of detection over a network, given a budgetary constraint.

The model was tested using two separate objectives; the first is the Max *AvgCov* objective, which maximizes the average probability of detection over all nodes of the network, and the second is the Max *MinCov* objective, which maximizes the minimum probability of detection at any node in the network. Each of the two objectives was tested using Visual Basic for Applications (VBA) algorithms and Frontline Systems' Premium Solver Platform with the OpQuest solver engine (OQS).

Of the two VBA algorithms, *VBA-AvgCov* and *VBA-MinCov*, the *VBA-AvgCov* algorithm produced much better results. However the OptQuest solver was able to produce quicker and better results than both of the VBA algorithms.

Using the OptQuest solver, the Max *MinCov* objective produced good results under the larger budgetary constraint, but could not find a feasible solution under the smaller budgetary constraint. The Max *AvgCov* objective produced good results regardless of the budgetary constraint. Both objectives resulted in good sensor placement solutions for coverage maximization and are recommended for further research.

Military Application

Due to the ease of use of the model (it runs on Microsoft Excel) and its portability (it can run on any laptop with the needed software) the model could be used by military forces in the field to determine positioning of sensors for border and perimeter security. While Google Earth was used to create the network for this model, troops in the field may not have access to this software due to the lack of internet connectivity. Nonetheless, the network can be just as easily created with any software capable of displaying latitude and longitude coordinates.

V. Recommendations

Sensor Locations and Ranges

The model assumes multiple sensor types are placed together as packages in one location. This assumption creates fewer physical sensor locations making it easier and cheaper to secure the network than if each sensor is allowed its own location. However, each sensor has a different range and, under the stated assumption, the sensors with the longest range overwhelmingly influences the placement of sensors, i.e. the longer the range of a sensor, the more nodes it detects. So, in effect optimization is prioritized based on the range of each sensor, with the longest range having the greatest optimization priority.

In order to improve upon the present solution, each sensor type will need to be selected, at each node, independent of the other sensor types. This will require either a quadrupling of the number of variables, constraints, and calculations in the model, or a separate budgetary constraint for each sensor type. If each sensor type is given its own budgetary constraint, a simplified version of this model can be solved separately for each sensor type. This will likely result in many more nodes with at least one type of sensor and the cost of building the network infrastructure may increase significantly.

Sensor Fusion

The model assumes independence and combines all probabilities using this assumption. However, in an operational environment, the assumption of independence may not be justifiable. There may be correlation between sensors of different types as well as between sensors of the same type at different locations. There may also be a number of additional issues regarding the fusion of data between individual sensors, which will need to be considered and integrated into the model architecture.

Summary

This research has developed a proof-of-concept sensor placement model for border interdiction. The model allows each node to easily be turned on or off for sensor placement based on economic, political, and operational considerations. It also has the capability to degrade, or turn off, specific node to node interactions. This capability allows more accurate models of locations with many obstacles and/or elevation changes to be created. However, the model has not been tested with operational inputs and needs to be modified before it can be employed in an operational environment.

Appendix A

Beta Probability Distribution

The "beta [probability] distribution is a two-parameter family of continuous probability distributions defined on the interval [0, 1]" (8). By changing the parameters, α and β, the Beta distribution can exhibit an infinite number of density function shapes. (8; 35; 36:178,179)

Beta Probability Density Function (pdf):

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1} = \frac{1}{\mathrm{B}(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

Γ is the gamma function and B is the beta function

Beta Density Function Shapes:

- $\alpha < 1, \beta < 1$ is U-shaped
- $\alpha < 1, \beta \geq 1$ or $\alpha = 1, \beta > 1$ is strictly decreasing
  - $\alpha = 1, \beta > 1$ is strictly convex
  - $\alpha = 1, \beta = 2$ is a straight line
  - $\alpha = 1, 1 < \beta < 2$ is strictly concave
- $\alpha = 1, \beta = 2$ is the uniform distribution
- $\alpha = 1, \beta < 1$ or $\alpha > 1, \beta \leq 1$ is strictly increasing
  - $\alpha > 2, \beta = 1$ is strictly convex
  - $\alpha = 2, \beta = 1$ is a straight line
  - $1 < \alpha < 2, \beta = 1$ is strictly concave
- $\alpha > 1, \beta > 1$ is unimodal
- If $\alpha = \beta$ then the density function is symmetric about ½

Examples:

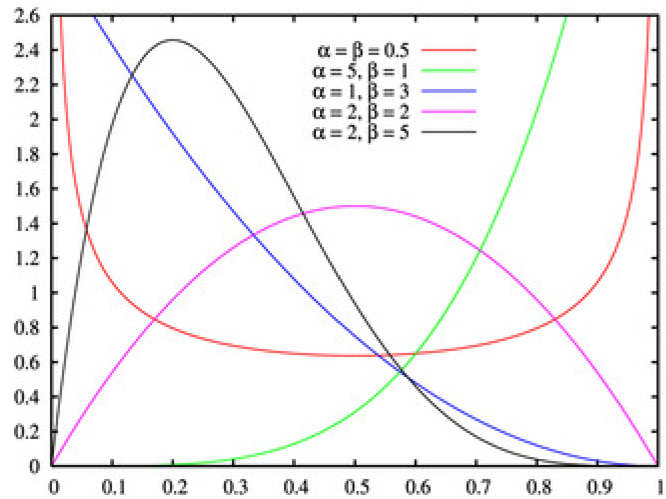


Figure 23. Probability density functions (8)

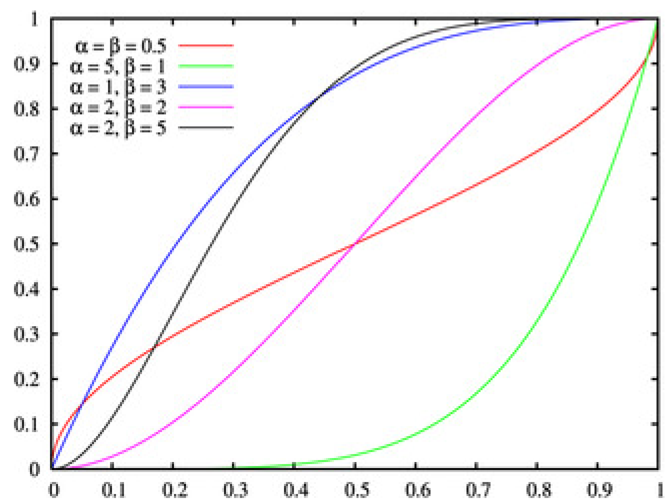


Figure 24. Cumulative distribution function (8)

Appendix B

The Visual Basic for Applications (VBA) code for the Max_Ave_Det and VBA-MinCov algorithms is presented below:

VBA-AvgCov Algorithm

```
Sub MyAveSolver()

    Dim MyCurNode As Range
    Set MyCurNode = Worksheets("IO").Range("T31")

    Dim MyFirstCell As Range
    Set MyFirstCell = Worksheets("IO").Range("E2")

    Dim MyLastCell As Range
    Set MyLastCell = Worksheets("IO").Range("E674")

    Dim MyTarCell As Range
    Set MyTarCell = Worksheets("IO").Range("T19")

    Dim MyBudget As Range
    Set MyBudget = Worksheets("IO").Range("N13")

    Dim MyCost As Range
    Set MyCost = Worksheets("IO").Range("N23")

    Dim MyPrev As Range
    Set MyPrev = Worksheets("IO").Range("T29")

    Dim MyNext As Range
    Set MyNext = Worksheets("IO").Range("T30")

    Dim MyTotal As Integer
    MyTotal = 0

    Application.Calculation = xlManual
    Do
        MyFirstCell.Offset(MyTotal, 0) = 0
        MyTotal = MyTotal + 1
    Loop Until MyFirstCell.Offset(MyTotal, 0).Address = MyLastCell.Offset(1, 0).Address
    Application.Calculation = xlAutomatic

    Dim MyCount As Integer
    Dim MyBestNode As Integer
```

```
    Dim MyBestValue As Double

    MyPrev.Value = 0
    MyNext.Value = 0

    Do
        MyBestNode = 0
        MyBestValue = MyTarCell.Value

        For MyCount = 1 To MyTotal

            MyCurNode = MyCount
            If ((MyFirstCell.Offset(MyCount - 1, 0) = 0) And (MyFirstCell.Offset(MyCount - 1, -1)
= 1)) Then
                MyFirstCell.Offset(MyCount - 1, 0) = 1
                If MyTarCell.Value > MyBestValue Then
                    MyBestNode = MyCount
                    MyBestValue = MyTarCell.Value
                End If
                MyFirstCell.Offset(MyCount - 1, 0) = 0
            End If
        Next MyCount

        MyFirstCell.Offset(MyBestNode - 1, 0) = 1

        MyPrev.Value = MyNext.Value
        MyNext.Value = MyBestNode

    Loop Until MyBudget.Value < MyCost.Value

    MyFirstCell.Offset(MyBestNode - 1, 0) = 0

End Sub 'End MyAveSolver
```

VBA-MinCov Algorithm

```
Sub MyMinSolver()

    Dim MyCurNode As Range
    Set MyCurNode = Worksheets("IO").Range("T31")

    Dim MyFirstCell As Range
    Set MyFirstCell = Worksheets("IO").Range("E2")

    Dim MyLastCell As Range
    Set MyLastCell = Worksheets("IO").Range("E674")
```

```
Dim MyTarCell As Range
Set MyTarCell = Worksheets("IO").Range("T19")

Dim MyTarMinCell As Range
Set MyTarMinCell = Worksheets("IO").Range("T20")

Dim MyBudget As Range
Set MyBudget = Worksheets("IO").Range("N13")

Dim MyCost As Range
Set MyCost = Worksheets("IO").Range("N23")

Dim MyPrev As Range
Set MyPrev = Worksheets("IO").Range("T29")

Dim MyNext As Range
Set MyNext = Worksheets("IO").Range("T30")

Dim MyTotal As Integer
MyTotal = 0

Application.Calculation = xlManual
Do
   MyFirstCell.Offset(MyTotal, 0) = 0
   MyTotal = MyTotal + 1
Loop Until MyFirstCell.Offset(MyTotal, 0).Address = MyLastCell.Offset(1, 0).Address
Application.Calculation = xlAutomatic

Dim MyCount As Integer
Dim MyBestNode As Integer
Dim MyBestValue As Double
Dim MyBestMinNode As Integer
Dim MyBestMinValue As Double


MyPrev.Value = 0
MyNext.Value = 0

Do
   MyBestNode = 0
   MyBestMinNode = 0

   MyBestValue = MyTarCell.Value
   MyBestMinValue = MyTarMinCell.Value
```

```
        For MyCount = 1 To MyTotal

            MyCurNode = MyCount
            If ((MyFirstCell.Offset(MyCount - 1, 0) = 0) And (MyFirstCell.Offset(MyCount - 1, -1)
= 1)) Then

                MyFirstCell.Offset(MyCount - 1, 0) = 1

                If MyTarCell.Value > MyBestValue Then
                    MyBestNode = MyCount
                    MyBestValue = MyTarCell.Value
                End If

                If MyTarMinCell.Value > MyBestMinValue Then
                    MyBestMinNode = MyCount
                    MyBestMinValue = MyTarMinCell.Value
                End If

                MyFirstCell.Offset(MyCount - 1, 0) = 0
            End If

        Next MyCount

        If MyBestMinValue = MyTarMinCell.Value Then
            MyFirstCell.Offset(MyBestNode - 1, 0) = 1
            MyPrev.Value = MyNext.Value
            MyNext.Value = MyBestNode
        Else
            MyFirstCell.Offset(MyBestMinNode - 1, 0) = 1
            MyPrev.Value = MyNext.Value
            MyNext.Value = MyBestMinNode
        End If

    Loop Until MyBudget.Value < MyCost.Value

        MyFirstCell.Offset(MyBestNode - 1, 0) = 0
        MyFirstCell.Offset(MyBestMinNode - 1, 0) = 0

End Sub 'End MyMinSolver
```

Appendix C

The solver input parameters are presented below. All four screen which allow user input or selection are shown.

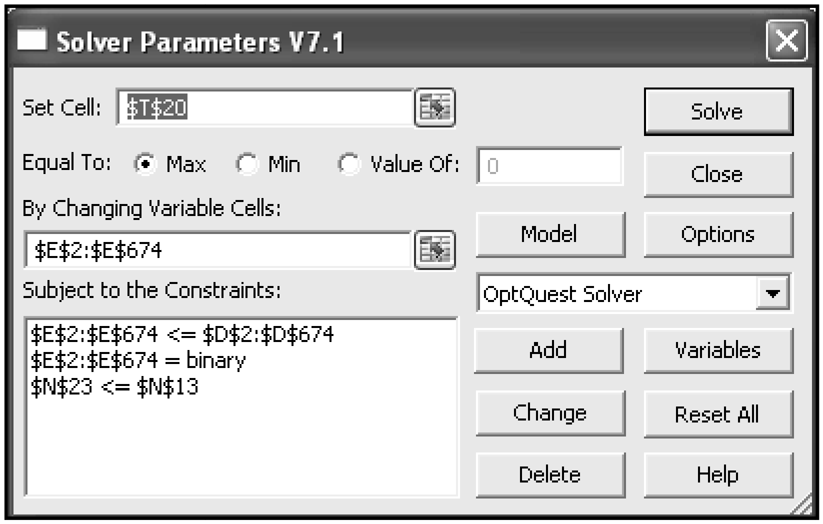


Figure 25. Solver Objective, Variable, and Constraint Input Screen

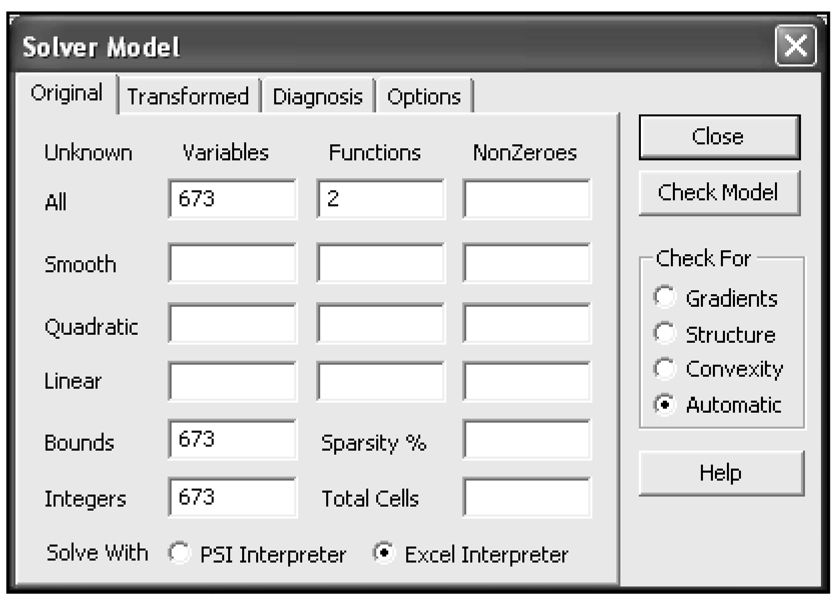


Figure 26. Solver Interpreter Selection Screen

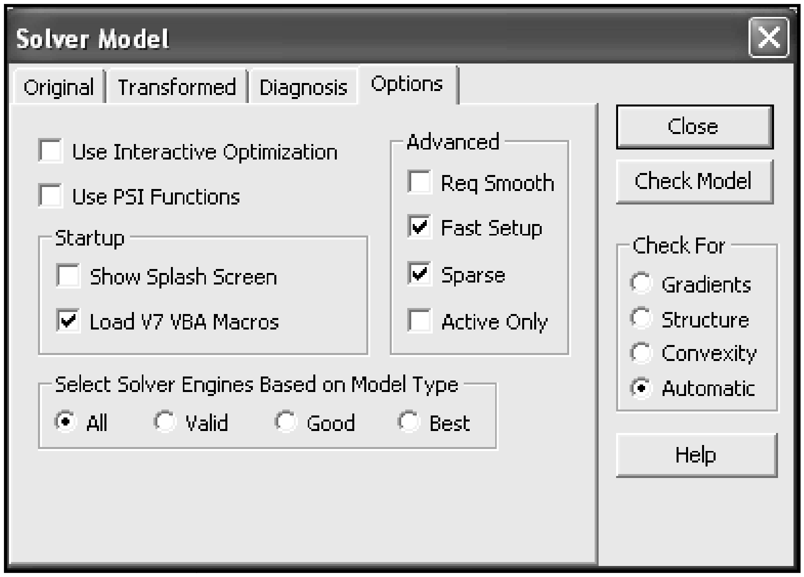


Figure 27. General Solver Options Selection Screen

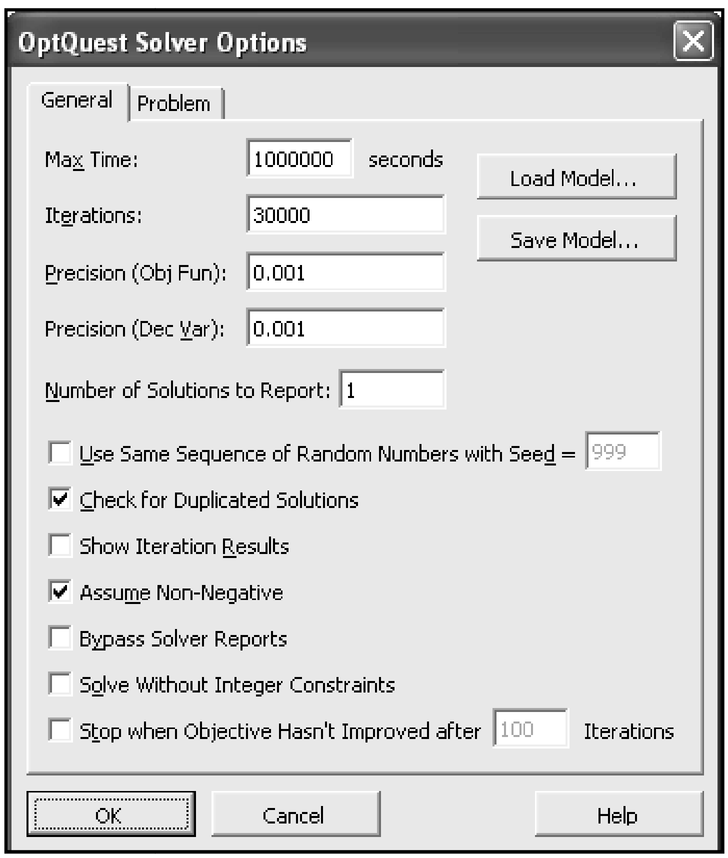


Figure 28. OptQuest Solver Options Input Screen

References:

1. Extreme CCTV Surveillance Systems, “GVS1000 long range Day/Night PTZ surveillance system,” (1 January 2007). 23 April 2007, http://www.extremecctv.com/product_detail.php?product_id=151

2. Axsys Technologies, “Thermal cameras - infrared cameras,” (2007). 22 April 2007, http://www.axsys.com/p-thermcam.php

3. Dragoon Technologies, “MSR-20 ground surveillance radar,” (2007). 9 March 2007, http://www.dragoontech.com/Media/DragoonBrochures/msr20.pdf

4. Dragoon Technologies, “Palm PDA based intelligence distribution,” (2007). 9 March 2007, http://www.dragoontech.com/PDF_Brochures/palm.pdf

5. U.S. Government Accountability Office, “BORDER PATROL: Available data on interior checkpoint suggest differences in sector performance.” U.S. Government Accountability Office, GAO-05-435, Washington D.C. (July 2005). 1 March 2007, http://www.gao.gov/new.items/d05435.pdf

6. Google Earth developers, “Google Earth (Release 4),” Google Inc. (8 January 2007). 20 February 2007, http://earth.google.com/earth4.html

7. Sgrillo, R., “GE-Path,” Google Earth Tools (14 June 2006). 20 February 2007, http://www.sgrillo.net/googleearth/

8. Wikipedia contributors. “Beta distribution,” Wikipedia – The Free Encyclopedia (December 2006). 22 February 2007, http://en.wikipedia.org/wiki/beta_distribution?oldid=108604864

9. Friedman, M., “U.S. population to reach 300 million this month,” USINFO (3 October 2006). 12 March 2007, http://usinfo.state.gov/xarchives/display.html?p=washfile-english&y=2006&m=October&x=20061003162245jmnamdeirf0.4331629

10. Konrad, Rachel, “Immigrants behind 25 percent of tech startups,” Associated Press. (3 January 2007). 20 February 2007, http://www.msnbc.msn.com/id/16459952/

11. King, D. A., “Could there be twenty million illegals in the U.S.?,” V-Dare (7 August 2004). 12 March 2007, http://www.vdare.com/king/illegals.htm

12. McCain, John, “Senate hearing reveals massive invasion.” CNN – Lou Dobs Tonight transcript (17 June 2004). 12 March 2007 http://www.americanpatrol.com/_WEB2004/040618.html

13. The Office of Border Patrol and The Office of Policy and Planning, “National border patrol strategy,” U.S. Customs and Border Protection, Washington D.C. (September 2004). 16 October 2006,

http://www.cbp.gov/linkhandler/cgov/border_security/border_patrol/national_bp_strategy.ctt/national_bp_strategy.pdf

14. Caldwell, Alicia A., “Texas-Mexico border standoff reported,” Associated Press, (25 January 2005). 9 March 2007, http://www.breitbart.com/article.php?id=D8FBOSCO3&show_article=1

15. World Net Daily writers, “Armed standoff on the Rio Grande.” Wordnetdaily.com Inc., (20 November 2005). 9 March 2007, http://www.worldnetdaily.com/news/article.asp?ARTICLE_ID=47497

16. World Net Daily writers, “Mexican troops defiant: 'border? what border?',” Wordnetdaily.com Inc., (2006, January 15). 9 March 2007, http://www.worldnetdaily.com/news/article.asp?ARTICLE_ID=48357

17. U.S. Customs and Border Protection, “CPB Border Patrol Sectors,” U.S. Customs and Border Protection, (2007). 20 February 2007, http://www.cbp.gov/xp/cgov/border_security/border_patrol/border_patrol_sectors/

18. Marries, Dan, ‘Technology on the border; is enough being used?,” KOLD News 13, (10 November 2005). 13 March 2007, http://www.kold.com/Global/story.asp?S=4033882

19. Olavsrud, Thor, “Microsoft's battle lines shifting to office?,” BostonInternet.com, (21 November 2002). 13 March 2007, http://boston.internet.com/news/article.php/1546531

20. Israeli, Etan, and R. Kevin Wood, "Shortest-Path Network Interdiction," *Networks,* 40:97-111 (2002).

21. Pulat, Halil. *A two-sided optimization of border patrol interdiction.* Master’s Thesis, Naval Postgraduate School (NPS), Monterey CA, June 2005 (ADA435596). 16 October 2006, http://handle.dtic.mil/100.2/ADA435596

22. Washburn, Alan. *Continuous network interdiction.* Technical Report, Naval Postgraduate School (NPS), Monterey CA, June 2006 (NPS-OR-007). 16 October 2006, http://handle.dtic.mil/100.2/ADA449515

23. Washburn, Alan, and Kevin Wood. “Two-person zero-sum games for network interdiction,” *Operations Research,* 43:243-251 (Mar/Apr 1995).

24. Dhillon, S. S., and K. Chakrabarty. *Sensor placement for effective coverage and surveillance in distributed sensor networks.* Report, Defense Advanced Research Agency, Arlington VA, 2003 (ADA445754). 23 March 2007, http://handle.dtic.mil/100.2/ADA445754

25. Dhillon, S. S., K. Chakrabarty, and S. S. Iyengar. *Sensor placement for grid coverage under imprecise detections.* Report, Defense Advanced Research Agency, Arlington VA, 2005 (ADA440417). 23 March 2007, http://handle.dtic.mil/100.2/ ADA440417

26. USBorderPatrol.com contributors. “US border at night,” USBorderPatrol.com (2007). 20 April 2007, http://www.usborderpatrol.com/Border_Patrol1604.htm

27. OptQuest Solver. “OptQuest solver engine for excel, premium solver platform.” Frontline Systems Inc. (2007). 23 February 2007, http://www.solver.com/xlsoptqeng.htm

28. Democratic Policy Committee. “Bush republicans fail to make border security a priority,” Democratic Policy Committee (23 May 2006). 10 March 2007 http://democrats.senate.gov/dpc/dpc-new.cfm?doc_name=fs-109-2-72

29. Gregory, David, and Norah O’Donnell, “Bush ordering guard troops to border.” MSNBC News, (15 May 2006). 13 March 2007, http://www.msnbc.msn.com/id/12796688/

30. LaJeunesse, William, “Mixed reports on national guard border confrontation,” Fox News. (1 February 2007). 13 March 2007 http://www.foxnews.com/story/0,2933,248124,00.html

31. HQ AMC, “DISTCALC distance calculator,” Internal-use application, Headquarters Air Mobility Command (HQ AMC).

32. Weisstein, Eric W., "Great circle." MathWorld - A wolfram web resource, (21 November 2002). 20 March 2007, http://mathworld.wolfram.com/GreatCircle.html

33. Weisstein, Eric W., and J. Pintér, "Tabu search." MathWorld - A wolfram web resource, (2 November 2004). 20 March 2007, http://mathworld.wolfram.com/TabuSearch.html

34. Laguna, Manuel, and Rafael Martí, *Scatter Search : Methodology and Implementations in C.* Boston: Springer, 2003.

35. Weisstein, Eric W., "Beta distribution." MathWorld - A wolfram web resource. (29 September 2003). 20 March 2007, http://mathworld.wolfram.com/BetaDistribution.html

36. Devore, J. L., *Probability and Statistics for Engineering and the Sciences* (5th Edition). Pacific Grove, CA: Brooks/Cole Publishing Company, 1999.

37. Meier, Eric W., *Buried Fiberoptic Intrusion Sensor.* Master’s Thesis, Texas A&M University, May 2004. 24 April 2007, https://txspace.tamu.edu/bitstream/1969.1/425/1/etd-tamu-2004A-ELEN-Maier-1.pdf

38. Moran, Mark L., “Adaptive Seismic Sensor Networks,” Presentation, US Army Corps of Engineers – Engineer Research and Development Center, (June 2002). 24 April 2007, http://www.dtic.mil/ndia/2002mines/moran.pdf

REPORT DOCUMENTATION PAGE				*Form Approved* *OMB No. 074-0188*
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**				
1. REPORT DATE *(DD-MM-YYYY)* 04-26-2007	**2. REPORT TYPE** **Master's Thesis**			**3. DATES COVERED** *(From – To)* Jun 2006 – April 2007
4. TITLE AND SUBTITLE OPTIMIZING DISTRIBUTED SENSOR PLACEMENT FOR BORDER PATROL INTERDICTION USING MICROSOFT EXCEL				**5a. CONTRACT NUMBER**
				5b. GRANT NUMBER
				5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Patrascu, Adrian C.., Lt, USAF				**5d. PROJECT NUMBER**
				5e. TASK NUMBER
				5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765				**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GOR/ENS/07-21
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A				**10. SPONSOR/MONITOR'S ACRONYM(S)**
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				
13. SUPPLEMENTARY NOTES				

14. ABSTRACT

The purpose of this research was to develop an electronic sensor placement model for border security. A model was developed using Microsoft Excel, with some add-on capabilities, to optimize the placement of electronic sensors on a border network given a pre-determined budgetary constraint. The model is capable of handling multiple sensor types, which are placed together as packages, and allows for daytime, nighttime, or 24 hour operation of each sensor type. Additionally, each sensor can be assigned a specific range and detection probability curve within the given range. The model is capable of optimizing either average coverage, or minimum coverage, across the nodes of a network by selecting the nodes where sensor packages are to be placed.

Due to its simplicity and ability to run in Microsoft Excel, it is believed that the model developed in this research can also be used in a number of military applications where border security is necessary.

15. SUBJECT TERMS

Border Security, Interdiction, Linear Programming, Non-Linear Programming, Networks, Integer Programming, Border Patrol, Optimization, Operations Modeling

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** James Moore, Ph.D. (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4528; e-mail: James.Moore@afit.edu
U	U	U	UU	81	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18